清 华 大 学

综 合 论 文 训 练

题目：基于安全多方计算的物联网隐私保护机制研究

系　　别：计算机科学与技术系

专　　业：计算机科学与技术专业

姓　　名：洪璐

指导教师：蒋屹新　副教授

2011 年 6 月 13 日

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签　名：_____导师签名：_____日　期：_____

中文摘要

物联网是未来网络的发展趋势。在不久的将来，物联网将会成为人们每天生活中不可缺少的一部分。与此同时，人们每天的活动信息甚至隐私信息都会暴露在物联网中。所以物联网应用中的隐私保障是物联网能广泛应用的基本要求之一。

物联网感知层结点由于通信计算能力有限，不能直接使用已有的安全保护机制。物联网的应用大部分都是由多个互不信任的参与方参与协同完成的，这就涉及到了安全多方计算问题。本文以高效的全同态加密机制为基础，首次提出了基于全同态加密的的适用于物联网感知层的安全多方计算协议。

关键词：物联网　感知层 全同态加密　安全多方计算

ABSTRACT

As the trend of future Internet, the Internet of things will pervade peoples' daily life. At the meanwhile, since people's daily activities information will be collected by the Internet of Things, it would not be surprising that one of the requirements to ubiquitous applications of Internet of Things is privacy protecting.

With limited computation and communication power, nodes in the perception layer of the Internet of Things cannot afford to use the existing privacy protecting scheme. On the other hand, since the computations in Internet of things are often performed among inputs from untrusting parties, secure multi party computations may become a relevant approach to solve the problem. In the paper, we propose the first privacy-protecting scheme of perception layer of the Internet of Things, based on secure multi party computations with Gentry's fully homomorphic encryption.

Keywords: Internet of Things, Perception Layer, Fully Homomorphic Encryption, Secure Multi Party Computation

目 录

第1章 引 言

1.1 研究背景

随着计算机技术和网络技术的发展，人们已不仅仅满足于将成千上亿台电脑连成网络带来的巨大使用价值。将身边的任何设备，包括手机、电视、空调等等，连成智能网络将是未来网络发展的目标。这就是物联网最初始的想法：把计算能力和通信技术嵌入到身边的任何设备中，通过这些设备组成网络而实现不同的应用。

物联网（Internet of Things，IoT），从字面上来理解就是“物与物相互连接的互联网”，其核心和基础仍是互联网，是在互联网基础上的延伸和扩展的网络，但其客户端延伸到了物体与物体之间、人与物体之间等等。

和传统网络对比，物联网有以下三个主要特征：

1. 它利用感知结点来实现与物理世界的融合。物联网的最底层是感知层，在感知层是由许多传感器结点构成的网络，每个传感器从物理世界中周期性地获取信息，向周围结点或者上层发送信息以及并交换信息。
2. 它是基于互联网的网络。物联网的基础与核心依然是互联网。物联网通过有线或者无线机制来与互联网融合，将从物理世界收集到的信息与互联网进行交互。
3. 物联网的最底层物理层不仅有传感器结点，也有控制功能，能够智能处理不同传感器结点间的协调等问题。在物联网的上层，利用云计算和模式识别等智能手段，与互联网应用相结合，具有更加广泛的应用前景。

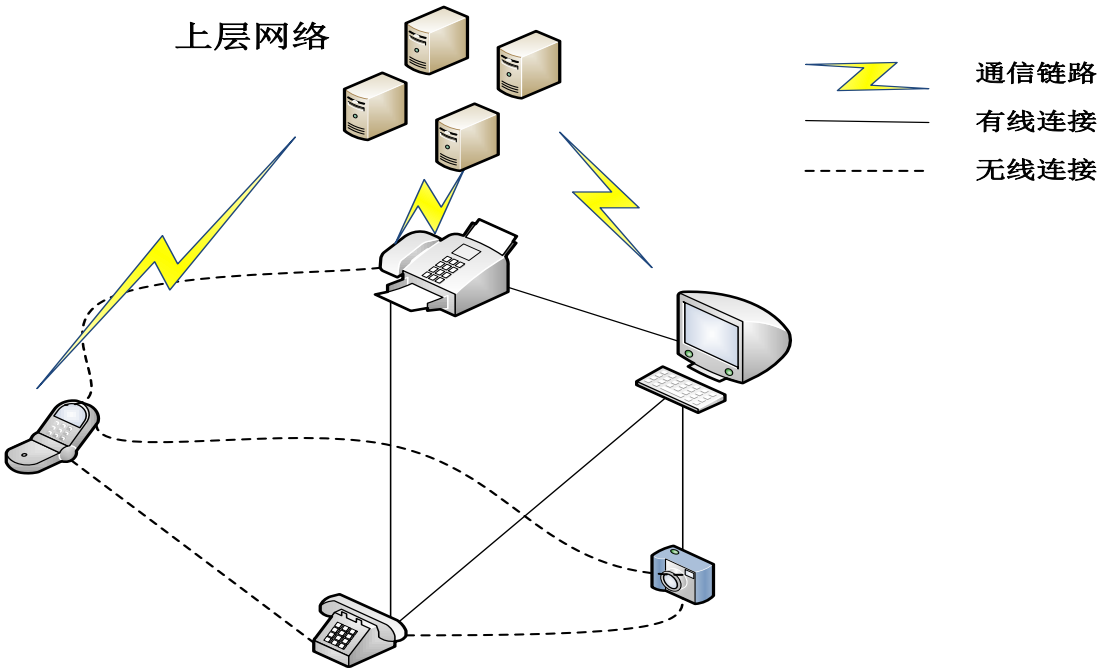

图1.1　物联网示意图

在不久的将来，物联网将会成为人们每天的生活中不可缺少的一部分。与此同时，人们每天的活动信息甚至隐私信息都会暴露在物联网中，例如行程、喜好等等。因此，在物联网中对数据进行加密是必要的。

物联网带来的诸多应用毫无疑问将会极大地提高人们的生活水平。这些应用的最大价值在于能让多个实体（即物联网中的“物”）进行合作、交互以及计算。然而，由于上述的交互以及运算大多数发生在互不信任的实体中，且运算中不可避免地包含着敏感的隐私数据，例如交通控制、健康信息维护以及军事场合等等。所以，如何在物联网多方交互过程中保证数据隐私性更是不容忽视的。

在物联网的最底层，也就是与现实的物理世界交互的感知层，负责从物理世界收集信息。它是一个由无线传感器构成的网络，其传感器节点的计算通信能力以及存储能力有限，无法直接使用跳频通信或者公钥密码等传统安全机制[1]。但由于其直接与物理世界交互，且节点交多，极易受到攻击而导致信息的泄露或者网络的瘫痪。所以，如何在保持物联网感知层数据隐私的同时减小算法复杂度提高处理效率是必须解决的安全问题。

安全多方计算可进行多个结点对敏感数据的计算，适用于物联网感知层模型。而由 Gentry 提出的全同态加密算法则为安全多方计算提供了一条高效安全的解

决方案。本论文将从全同态加密与安全多方计算入手，研究物联网感知层的隐私保护机制。

1.2 论文的研究内容

本论文的主要工作是对物联网感知层的隐私保护机制进行了研究与探索。物联网是未来网络的核心，其最低层感知层的隐私保护十分重要。本论文的主要研究内容有：

1. 归纳了物联网及其在隐私保护要求下的相关背景与问题。
2. 深入研究了 Gentry 全同态加密算法以及现有的安全多方解决方案。
3. 提出了物联网感知层安全模型。本文基于安全多方计算并结合 Gentry 全同态加密算法，设计出了一个可适用于物联网感知层的隐私保护机制，并对其进行了改进。

本论文的研究内容用图表示如下：

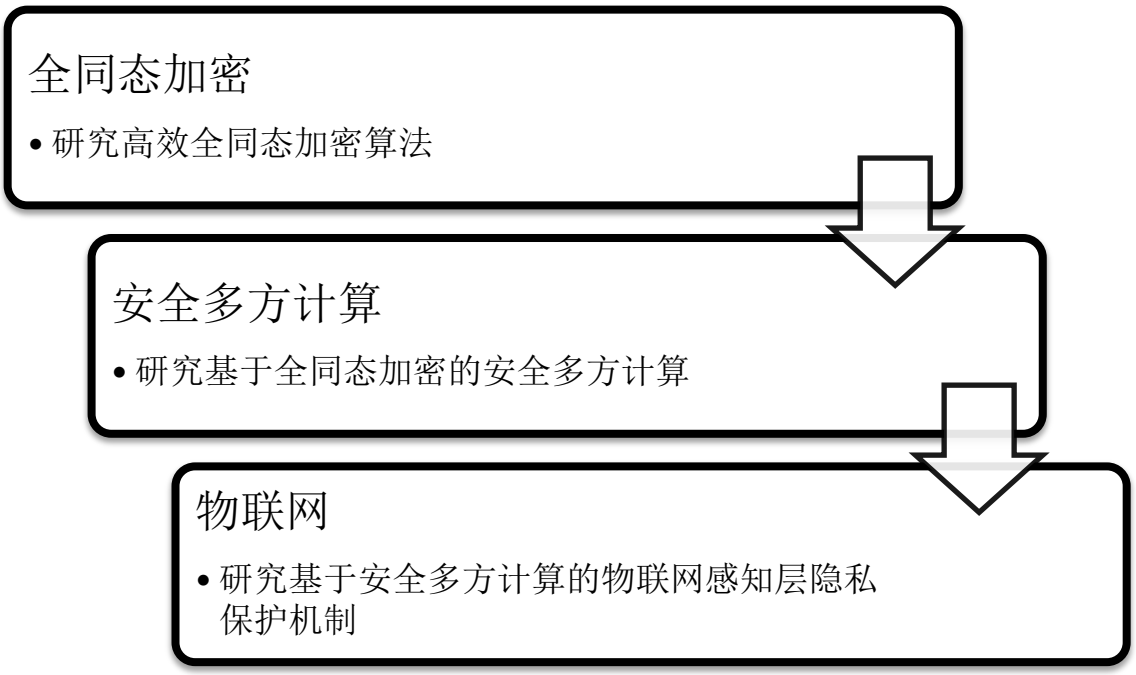

图1.2　论文研究内容

1.3 论文的组织结构

本论文共分为章，每章主要内容如下：

第一章主要介绍了物联网及其在隐私保护要求下的相关背景与问题，并介绍了本论文研究的主要内容以及组织结构。

第二章介绍了全同态加密的研究背景与研究现状，深入研究了 Gentry 全同态加密机制，归纳了其研究的关键技术并分析了将其应用在物联网感知层的可行性。

第三章介绍了安全多方计算的研究背景与研究现状。

第四章基于安全多方计算并结合 Gentry 全同态加密算法，设计出了一个可适用于物联网感知层的隐私保护机制，并对其进行了改进。

1.4 本章小结

本章对物联网隐私保护及其研究背景进行了简要介绍，并介绍了本论文的主要研究内容以及本论文的组织结构。

第2章 全同态加密

2.1 基本概念

自古以来，加密的目标就是保证数据在传输和存储过程的安全性。近年来，信息技术的发展对加密提出更多的要求。例如，用户把数据放到具有较强计算能力但是不信任的服务器上进行处理，为了保障数据的安全，就必须先对数据进行加密。服务器只能对已加密的数据进行处理，当处理完毕，服务器把结果返回给用户，用户通过解密得到自己想要的结果。在此过程中，服务器无法得知与原数据相关的任何有用信息。上述例子就是同态加密的一个具体应用。

同态加密可以在不同的物联网应用加以使用。

在农业中，比如在一个农业地区，有好几个农场内都置有许多感应器来收集地里的农药信息，当地农业局想收集一下当地农场农药使用的平均浓度，但农场主却不希望别人知道自己农场的农药浓度，这就是农场主想要保护的敏感数据。这时可以把各农场的农药浓度进行同态加密后再进行计算，就可以得到所有农场农药使用的平均浓度，而不泄露每个农场的农药深度了。

在野生动物保护中，科学家为了实时监测野生动物的情况，可以在野生动物的身上安装一个小型传感器，来搜集动物的位置信息、身体状况等以了解某地区的野生动物生存情况。为了防止这些信息被狩猎者或者不法分子获取，对野生动物带来威胁，这些信息必须进行加密后才能进行传输。如果使用的是同态加密，那么收集到信息后，就可以直接进行计算得到汇总的信息而不泄露每个野生动物单独的信息。

简单来说，同态加密就是一类特殊的加密方法，在不知道解密函数的情况下，对密文进行的一些操作等效于对相应的明文进行一定的操作，即操作后的密文进行解密后，是对原明文进行一定操作后的新的有意义的明文。

同态加密的数学定义如下：

对任意给定的密钥 k，明文空间 M，密文空间 C，加密函数 E 满足

$$\forall m_1, m_2 \in M, E(m_1 \oplus_M m_2) = E(m_1) \oplus_C E(m_2)$$

则称该加密方案为同态加密方案，其中 $\oplus_M$ 和 $\oplus_C$ 分别是 M 和 C 上的某些操作。

如果 $\oplus_M$ 和 $\oplus_C$ 是加法运算，则称为加同态。

如果 $\oplus_M$ 和 $\oplus_C$ 是乘法运算，则称为乘同态。

如果一个方案既是加同态又是乘同态的，那么就称为全同态。

2.2 研究现状

RSA[2]是第一个同态加密机制：给定密文 $c_1 = p_1^e \bmod N$ 和 $c_2 = p_2^e \bmod N$，可以计算 $c = c_1 \cdot c_2 = (p_1 \cdot p_2)^e \bmod N$，c 是原明文乘积的密文。然而，该机制是确定型的，从而不能满足语义安全的要求。尽管如此，但 RSA 的乘法同态性仍得到了广泛的应用。

Rivest, Adleman 和 Dertouzos[3]首次提出了隐私同态的概念，也提出了许多可能的机制，但最终都被证明是有缺陷的[4]。

是否能保证语义安全是同态加密的一个关键考虑因素。Boneh 和 Lipton 证明了任意基于环 $\mathbb{Z}_n$ 的代数同态加密机制如果是确定性的，那么它就能在低于指数时间内被攻破[5]。

第一个提出具有语义安全的同态加密机制是 Goldwasser-Micali[6]。其它的一些具有语义安全的加同态机制有 Benaloh[7]，Naccache-Stern[8]，Okamoto-Uchiyama[9]，Paillier[10]等等。ElGamal[11]是具有乘同态的机制。Bonech-Goh-Nissim[12]机制同时具有加同态和乘同态性质，但它只允许对密文进行二次方程操作。由 Fellows 和 Koblitz[13]提出的“Polly Cracker”算法虽然能对任意电路进行计算，但是密文规模会随着电路深度的增加呈指数型增长。还有许多其它的机制，如 Sander[14]以及 Ly[15]提出的机制，均有密文规模指数爆炸的缺陷。

2.3 Gentry 的全同态加密

2009 年，IBM 的 Gentry[16]提出了基于理想格的全同态加密，解决了上述全同态加密方案中存在的问题，能对任意电路进行有效计算，而且不存在密文规模指数爆炸的问题。

其方案的关键思想主要有三点：

1. 电路的可自启性。即当方案可以同态计算其解密电路以及解密电路通过不同的门进行组合的电路。这一性质为该方案能对任意电路进行有效同态运算提供了保障。
2. 理想格的利用。理想格中的运算具有同态性，而传统的基于格的计算难题也保障了安全性。
3. 压缩解密电路。通过减小解密电路的复杂度，来提高方案所能计算的电路深度。

下面就以上面三点来简要介绍这一全同态加密方案。

2.3.1 可自启性（Bootstrappable）

假设已经有一个加密机制ε，它只能对某些特定的电路集C_ε进行同态操作，而且C_ε包含ε的解密电路D_ε。我们希望能通过ε构建一个能对任意电路进行同态操作的全同态加密机制。为了达到这点，先通过ε建立一个能对电路深度至多为 d 的任意电路进行同态操作的同态加密机制$\varepsilon^{(d)}$。$\varepsilon^{(d)}$的解密电路仍然是D_ε，私钥和密文的大小仍和ε中的一样。$\varepsilon^{(d)}$的公钥包含 d+1 个来自于ε的公钥，以及一条ε的私钥加密链，即第一个ε的私钥被第二个ε公钥加密，如此以往。对任意整数 d，集合$\{\varepsilon^{(d)}\}$被称为部分全同态加密机制，因为它的公钥大小与电路深度 d 有关。那么如何把一个部分全同态加密机制变为正的全同态加密机制呢？只需要在公钥中设一个循环即可，这样公钥大小就与电路深度无关了。

下面先给出一些与自启性有关的定义。

连接型解密电路：设Γ为输入和输出均在明文空间 P 中的集合。对任意门$g \in \Gamma$，若一个电路由一个 g 门并联连接多个D_ε（D_ε的数量为 g 门的输入端数），则称该电路为由 g 门连接的解密电路，这样的电路集合记为$D_\varepsilon(\Gamma)$。

可自启性：设机制ε对电路集C_ε是同态的。如果$D_\varepsilon(\Gamma) \subseteq C_\varepsilon$，则称ε对Γ是可自启的。

设$\mathrm{D}_\varepsilon(\Gamma,\delta)$表示的是深度为$\delta$，使用Γ中的门来连接$\mathrm{D}_\varepsilon$的电路集，即$\mathrm{D}_\varepsilon$成为这个深度为$\delta$的电路的输入。设ε对Γ是可自启的。对任意整数对$\mathrm{d} \geq 1$，使用ε来构造机制$\varepsilon^{(d)} = (KeyGen_{\varepsilon^{(d)}}, Encrypt_{\varepsilon^{(d)}}, Evaluate_{\varepsilon^{(d)}}, Decrypt_{\varepsilon^{(d)}})$，该机制能处理Γ中所有深度为 d 的电路。其具体算法如下：

- $\boldsymbol{KeyGen}_{\varepsilon^{(d)}}(\lambda, d)$

以秘密参数λ和正整数 d 为输入。$\ell = \ell(\lambda)$为明文和私钥长度。进下下面两步操作：

$(sk_i, pk_i) \overset{R}{\leftarrow} KeyGen_\varepsilon(\lambda), \mathrm{i} \in [0, \mathrm{d}]$

$\overline{sk_{ij}} \overset{R}{\leftarrow} Encrypt_\varepsilon\left(pk_{i-1}, sk_{ij}\right), \mathrm{i} \in [1, \mathrm{d}], \mathrm{j} \in [1, \ell]$

其中sk_{ij}是sk_i的第 j 位。输出私钥$sk^{(d)} = sk_0$，公钥$pk^{(d)} = \left(\langle pk_\mathrm{i}\rangle, \langle\overline{sk_{ij}}\rangle\right)$。设$\varepsilon^{(\delta)}$表示使用$sk^{(\delta)} = sk_0$，$pk^{(\delta)} = \left(\langle pk_\mathrm{i}\rangle_{\mathrm{i}\in[0,\delta]}, \langle\overline{sk_{ij}}\rangle_{\mathrm{i}\in[0,\delta]}\right), \delta \le \mathrm{d}$的子系统。

- $\boldsymbol{Encrypt}_{\boldsymbol{\varepsilon}^{(d)}}(pk^{(d)}, \pi)$

  输入为公钥$pk^{(d)}$和明文$\pi \in \mathrm{P}$。输出密文$\psi \overset{R}{\leftarrow} \boldsymbol{Encrypt}_{\boldsymbol{\varepsilon}}(pk_d, \pi)$。

- $\boldsymbol{Decrypt}_{\boldsymbol{\varepsilon}^{(d)}}(sk^{(d)}, \psi)$

  输入为私钥$sk^{(d)}$和密文ψ（在公钥pk_0下加密）。输出$\boldsymbol{Decrypt}_{\boldsymbol{\varepsilon}}(sk_0, \psi)$。

- $\boldsymbol{Evaluate}_{\boldsymbol{\varepsilon}^{(\delta)}}(pk^{(\delta)}, C_\delta, \Psi_\delta)$

  输入为公钥$pk^{(\delta)}$，由Γ中的门连接的深度至多为δ的电路C_δ，以及一组输入的密文Ψ_δ（这些密文均在公钥pk_δ下加密）。如果$\delta = 0$，则输出Ψ_0，否则执行下面三步操作：

  a. $(C^*_{\delta-1}, \Psi^*_{\delta-1}) \leftarrow \boldsymbol{Augment}_{\boldsymbol{\varepsilon}^{(\delta)}}\left(pk^{(\delta)}, C_\delta, \Psi_\delta\right)$

  b. $\left(C_{\delta-1}, \Psi_{\delta-1}\right) \leftarrow \boldsymbol{Reduce}_{\boldsymbol{\varepsilon}^{(\delta-1)}}\left(pk^{(\delta-1)}, C^*_{\delta-1}, \Psi^*_{\delta-1}\right)$

  c. $\boldsymbol{Evaluate}_{\boldsymbol{\varepsilon}^{(\delta-1)}}(pk^{(\delta-1)}, C_{\delta-1}, \Psi_{\delta-1})$

- $\boldsymbol{Augment}_{\boldsymbol{\varepsilon}^{(\delta)}}(pk^{(\delta)}, C_\delta, \Psi_\delta)$

  输入为公钥$pk^{(\delta)}$，由Γ中的门连接的深度至多为δ的电路C_δ，以及一组输入的密文Ψ_δ（这些密文均在公钥pk_δ下加密）。它用D_ε来增强C_δ，得到新电路$\mathrm{C}^*_{\delta-1}$。$\Psi^*_{\delta-1}$是新的密文集，通过把每个输入密文$\psi \in \Psi_\mathrm{d}$用$(\langle\overline{sk_{\delta j}}\rangle, \langle\overline{\psi_j}\rangle)$来代替，其中$\overline{\psi_j} \leftarrow \boldsymbol{Encrypt}_{\boldsymbol{\varepsilon}^{(\delta-1)}}(pk^{(\delta-1)}, \psi_j)$。

- $\boldsymbol{Reduce}_{\boldsymbol{\varepsilon}^{(\delta)}}\left(pk^{(\delta)}, C^*_\delta, \Psi^*_\delta\right)$

  输入为公钥$pk^{(\delta)}$，密文集Ψ^*_δ（每个密文都必须是$\mathrm{Encrypt}_{\varepsilon^{(\delta)}}$的像），电路$C^*_\delta \in D_\varepsilon(\Gamma, \delta + 1)$。它置$C_\delta$为$C^*_\delta$的子电路，包含前$\delta$级。置$\Psi_\delta$为$C_\delta$产生的输入密文，其中与线路 w 相关的密文$\psi^{(w)}_\delta$被置为$\boldsymbol{Evaluate}_{\boldsymbol{\varepsilon}}(pk_\delta, C^{(w)}_\delta, \Psi^{(w)}_\delta)$，其中$C^{(w)}_\delta$是输出为线路 w 的$C^*_\delta$的子电路，$\Psi^{(w)}_\delta$是$C^{(w)}_\delta$的输入密文。

2.3.2 理想格(Ideal Lattice)

以往的全同态加密机制都关注于如何最大化能同态计算的复杂度，而 Gentry 机制则关注于如何最小化解密电路的复杂度。它通过使用理想格来构造全同态加

密制，以达到最小化解密电路复杂度的目的，因为在基于格的密码系统的解密电路一般是由某些简单的操作构成，例如很容易并行化的矩阵乘法操作。下面来简要介绍一下利用理想格构造的适用于浅电路的同态加密机制。把该机制做为上一节中的ε机制，就可以得到一个原始的全同态加密机制了。

设 R 是环，I 是 R 的理想，B_I 是 I 的基，该方案由以下几个算法组成：

- $\boldsymbol{IdealGen}(R, B_I)$：输入为环 R 和 R 中理想 I 的基 B_I。输出基于某理想 J 的公有基 B_J^{pk} 和私有基 B_J^{sk}，其中 J 使得 $I+J=R$。
- $\boldsymbol{Samp}(B_I, x)$：输入为基 B_I，$x \in R$。输出为陪集 $x+I$ 中的随机抽样值。
- $\boldsymbol{KeyGen}(\mathrm{R}, \mathrm{B_I})$：输入为环 R 和 I 的基 $\mathrm{B_I}$。它置 $(B_J^{sk}, B_J^{pk}) \xleftarrow{R} \boldsymbol{IdealGen}(R, B_I)$。明文空间 P 是 $R \bmod B_I$ 的一个子集。公钥 $pk = (R, B_I, B_J^{pk}, \boldsymbol{Samp})$，私钥 $sk = B_J^{sk}$。
- $\boldsymbol{Encrypt}(pk, \pi)$：输入为公钥 pk 和明文 $\pi \in P$。它置 $\psi' \leftarrow \boldsymbol{Samp}(B_I, \pi)$，输出 $\psi \leftarrow \psi' \bmod B_J^{pk}$。
- $\boldsymbol{Decrypt}(sk, \psi)$：输入为私钥 sk 和密文 ψ。输出为明文 $\pi \leftarrow \left(\psi \bmod B_J^{sk}\right) \bmod B_I$。
- $\boldsymbol{Add}(pk, \psi_1, \psi_2)$：输出 $\psi_1 + \psi_2 \bmod B_J^{pk}$。
- $\boldsymbol{Mult}(pk, \psi_1, \psi_2)$：输出 $\psi_1 \times \psi_2 \bmod B_J^{pk}$。
- $\boldsymbol{Evaluate}(pk, C, \Psi)$：输入公钥 pk，输入密文集合Ψ以及在某一给定电路集 C_ε 中由 ***Add*** 门和 ***Mult*** 门构成的电路 C，可以用来计算函数 $f(\Pi)$ 的密文，其中Π为Ψ对应的明文集。它调用 ***Add*** 和 ***Mult*** 来进行计算。

在计算时，解密方程 $\pi \leftarrow \left(\psi \bmod B_J^{sk}\right) \bmod B_I$ 可以转化为 $\pi = \psi - B_J^{sk} \cdot [\left(B_J^{sk}\right)^{-1} \cdot \psi] \bmod B_I$。Gentry 通过把私钥改进为向量 $v_J^{sk}, v_J^{sk} \in J^{-1}$，以减小计算量，改进后的解密方程 $\pi = \psi - [v_J^{sk} \cdot \psi] \bmod B_I$。

2.3.3 压缩解密电路

在 Gentry 的方案中，压缩解密电路的思想是把原始的解密算法分成两部分，首先是一个繁重的不需要私钥的初始化预处理，由加密者完成，通过往公钥里添加一个与私钥有关的参数 τ 实现，其次是一个简单的需要私钥的处理，由解密者完成。

下面来简要介绍如何压缩解密电路。

设ε^*为初始加密机制，改进后的机制为ε，它使用两个新的算法$\boldsymbol{SplitKey_\varepsilon}$和$\boldsymbol{ExpandCT_\varepsilon}$。新机制由以下几个算法组成：

- $\boldsymbol{SplitKey_\varepsilon}(sk^*, pk^*)$：输入为原机制的私钥$sk^*$，从$sk^*$中得到密钥向量$\mathrm{v_j^{sk^*}}$。输出是新的私钥 sk 和与私钥有关的参数τ。其中τ为含有$\gamma_{setsize}(n)$个向量的向量集$t_1, \dots, t_{\gamma_{setsize}(n)}$，它们是$J^{-1}\ mod\ B_I$中的随机向量，并且存在一个基为$\gamma_{subsetsize}(n)$的子集$S \subset \{1, \dots, \gamma_{setsize}(n)\}$使得$\sum_{i\in S} t_i \in v_j^{sk^*} + I$。$sk$ 是一个$\gamma_{subsetsize}(n) \times \gamma_{setsize}(n)$的01矩阵M，其中当j是S的第i个元素时$M_{ij} = 1$。
- $\boldsymbol{ExpandCT_\varepsilon}(pk, \psi^*)$：输入为公钥 pk 和原方案产生的密文ψ^*。对$i \in [1, \gamma_{setsize}(n)]$，输出$\mathrm{c_i} \longleftarrow t_i \times \psi_i\ mod\ B_I$。
- $\boldsymbol{KeyGen_\varepsilon}(\lambda)$：输入为私密参数$\lambda$，输出为密钥对$(pk, sk)$。调用$(pk^*, sk^*) \overset{R}{\leftarrow} \boldsymbol{KeyGen_{\varepsilon^*}}(\lambda)$，以及$(sk, \tau) \overset{R}{\leftarrow} \boldsymbol{SplictKey_\varepsilon}(sk^*, pk^*)$。私钥是 sk，公钥是(pk^*, τ)。
- $\boldsymbol{Encrypt_\varepsilon}(pk, \pi)$：输入为公钥 pk 和明文π，输出为密文ψ。
  调用$\psi^* \leftarrow \boldsymbol{Encrypt_{\varepsilon^*}}(pk, \pi)$，并置$\psi$为$\psi^*$和$\boldsymbol{ExpandCT_\varepsilon}(pk, \psi^*)$的输出。
- $\boldsymbol{Decrypt_\varepsilon}(sk, \psi)$：输入为私钥 sk 和密文ψ，输出为明文π。进行以下操作：
  a. $\mathrm{w_{ij}} \leftarrow M_{ij} \cdot c_j$
  b. $\mathrm{x_i} \leftarrow \sum_{j=1}^{\gamma_{\mathrm{setsize}}(\mathrm{n})} w_{ij}$
  c. 从$x_1, \dots, x_k$中，其中$k = \gamma_{subsetsize}(n)$，构造 $k+1$ 个整向量$y_1, \dots, y_{k+1}$，使得其和为$[\sum_{i=1}^{\mathrm{k}} x_\mathrm{i}]$。
  d. $\pi \longleftarrow \psi - (\sum_{i=1}^{\mathrm{k+1}} y_\mathrm{i})\ mod\ B_I$

2.3.4 性能

对于安全性，Gentry 证明了他的方案是具有语义安全性的。并说明了其全同态加密机制的效率和安全性基于如下两个问题：

- 第一个是基于理想格的基本机制的安全性与效率。
- 第二个来自于向公钥中添加的与私钥有关的参数τ。

然而上述两个问题是互相制约的。增大$\gamma_{\mathrm{subset}}(n)$能增大第二个问题的强度，却以增大第一个问题中的约化因子，削弱其强度为代价。

粗略分析可知，在已知的攻击手段下，第二个问题的攻破时间约为$2^{\gamma_{\mathrm{subset}}(\mathrm{n})}$。第一个问题的约化因子也约为$2^{\gamma_{\mathrm{subset}}(\mathrm{n})}$。根据经验，一个约化因子为$2^{\mathrm{k}}$的格问题的攻破时间通常为$2^{\mathrm{n/k}}$，于是第一个问题的攻破时间约为$2^{\mathrm{n}/\gamma_{\mathrm{subset}}(n)}$。令

$\gamma_{\mathrm{subset}}(n) = \sqrt{n}$能最大化两个问题攻击所需的最小时间，为$2^{\sqrt{n}}$。为了使这一攻击时间为安全参数$\lambda$的指数，需要有$n \approx \lambda^2$。

Evaluate 算法的复杂度是λ的多项式。因为尽管电路深度很浅，但由于机制中私钥很长，所以导致算法复杂度只能达到λ的多项式。

在 Gentry 后来给出的可行版本中，它给出了在一个 32 位 CPU 的情况下，计算格维数为 512，公钥大小为 70MByte 的 Gentry 全同态加密机制需要 30 秒，而时间随着公钥大小的增长约为指数型增长。这一数字看起来并不是很理想，但在实际应用中，我们并不需要那么大的公钥和格维数，我们可以通过减少公钥大小来减少运算时间以及通信复杂度。

2.4 本章小结

本章首先介绍了全同态加密的基本概念，以及它在物联网中的应用背景。然后介绍了全同态加密的研究现状，最后详细介绍了 Gentry 全同态加密机制。该机制能对任意电路进行有效计算，而且不存在密文规模指数爆炸的问题，适用物联网感知层结点通信计算能力有效的情况。其关键思想主要有三点：是电路的可自启性、理想格的利用和压缩解密电路。

第3章 安全多方计算

3.1 基本概念

3.1.1 定义

安全多方计算(Secure Multiparty Computation)是指在无可信第三方的情况下，有 n 个参与方想要共同进行某种运算，如何在保证运算结果正确性的同时保证每个参与者数据隐私性的问题。它可以概括为如下的数学模型：

假设有 n 个参与者 1,…,n，他们分别持有隐私数据$x_1, \dots, x_n$。他们希望在不泄露$x_1, \dots, x_n$的同时，计算$f(x_1, \dots, x_n) = (y_1, \dots, y_n)$，使得参与方 i 只得到$y_i$而不知道其它输出信息[17]。

3.1.2 攻击者模型

在安全多方计算中，经常会考虑有攻击者，可能会腐化一些参与方或者攻击参与方的电脑。为了使协议能够正确执行，就需要考虑具有攻击者的情况。

按攻击者的主动性，可以分为：

- 被动攻击者

  被动攻击者只能得到被腐化的参与方所持有的信息，通过这些信息推测出有用信息，但是协议仍能正确地执行。

- 主动攻击者

  主动攻击者能完全控制被腐化的参与方，控制其发送的信息，破坏协议的执行。

按腐化的参与方的选定形式，可以分为：

- 非适应性的攻击者

  被腐化的参与方在协议开始前就已经选定，并且不能在协议进行中进行更改。

- 适应性的攻击者

  攻击者可以在协议进行时的任何时刻，根据自己已经掌握和需要的信息来选择腐化的对象。

3.1.3 通信模型

传统的安全多方计算有以下两种通信模型[18]：

- 密码学模型

  在这个模型中，攻击者能窃取所有发出的消息，但是不能篡改诚实参与方之间的交互的消息。这个模型中的参与方共享一个不安全的信道，安全性只能在密码学基础下保证，也就是说假设攻击者无法解决某些密码学计算难题。

- 信息论模型

  在信息论模型中，假设所有参与方都能在安全信道上交换信息，即攻击者无法得到任何诚实参与方之间交互的信息，即使攻击者有无限的计算能力也能保证安全性。

3.2 研究现状

安全多方计算起源于1982年Andrew Yao的百万富翁问题[19]：两个百万富翁想知道谁更富有，但他们不想让对方知道自己到底有多少资产。

1988年，Ben-Or, Goldwasser和Wigderson[20]以及Chaum, Crepeau和Damgard[21]证明了当存在一个自适应被动攻击者或者一个自适应主动攻击者时，若被攻击者收买的参与者少于n/2或n/3时，是存在安全计算方案的。

随后在1989年，Rabin和Ben-Or[22]提出了当存在广播信道、存在一个自适应主动攻击者以及被攻击者收买的参与者少于n/2时的安全计算方案。目前效率最高的协议是由Damgard和Nielsen[23]提出的，他们使用了线性秘密共享技术。

对于一般的攻击者，Hirt和Maurere[24]提出了攻击者腐化任意参与方的某一子集的方案。

Canetti, Lindell, Ostrovsky和Sahai[25]根据前人的工作，证明了即使有n-1个参与方被腐化，安全多方协议也可能达到一定的安全性，但这可能会导致协议失败，在腐化参与方得不到正确信息的同时诚实参与者也得不到任何信息。

3.3 本章小结

本章首先介绍了安全多方计算的定义、攻击者模型以及通信模型，并介绍了安全多方计算目前的研究现状。

第4章 基于安全多方计算的物联网感知层隐私保护协议

4.1 需要考虑的问题

4.1.1 结点的能力

感知层中的每个结点是否是平等的，计算能力和通信能力是否完全相同。有以下两种方案：

1. 各结点计算通信能力相同。这种方案相在现实中来说比较容易实现，因为不需要物联网感知层中特意寻找或者设置计算能力和通信能力较强的结点，而且由于每个结点都是平等的，部分结点的瘫痪并不会导致整个网络的瘫痪。
2. 有些结点计算通信能力相对较强。这种方案对协议的实现与效率有很大的帮助，因为如果存在计算和通信能力较强的结点，可以将大部分计算能力交给这样的结点，以提高效率。但缺点是这些结点由于成为了“出头鸟”，更加容易受到攻击，而这些结点如果瘫痪了就会导致整个网络的瘫痪。

在我们的协议中采用后一种方案，原因在于对于物联网感知层来说，大部分结点是计算通信能力有限的传感器，如果各结点通信能力都相同，那么很难进行复杂的运算。而对于计算通信能力强的结点，其抗攻击能力也会稍强，而且抗攻击力也与软件有关。同时可以设置多个这样的结点，分散攻击者的注意力。

4.1.2 密钥的生成

需要考虑的问题有密钥应该由谁生成，如何发布等。

协议涉及到多个感知层结点的计算，每次计算可能是所有结点都必须参与，也可能只需一部分结点参与，所以应该考虑每次计算是否都应该生成新的密钥，由哪个结点生成密钥。

有以下两种解决方案：

1. 固定一个或少数几个通信能力和计算能力较强的结点生成密钥，定期更换。这种方案相对来说要简单易行，但对抗攻击能力较弱。
2. 密钥共享。由一个结点生成密钥，并将它共享到 n 个结点中，只要这 n 个结点中的 t 个提供了正确的密钥信息才能将密钥恢复。这一方案对攻击者存在的情况下有较好的抵抗能力。

在我们的协议中，采用第一种方案，主要原因是这种方案比较简单易行。可以将第二个方案作为改进的努力方向。

4.1.3 计算的承担

需要考虑的是计算应该由哪些结点承担，如何在计算时保障安全性。

有以下两种方案：

1. 计算由多个结点承担。一个结点计算一部分后发给下一结点接着计算。优点是将计算任务分给多个结点，每个结点的任务就很轻了，不需要有计算能力较强的结点。缺点是需要考虑复杂的传递顺序方案，计算中的某一环节的一个结点若发生问题，如被攻破或者变成隐藏攻击者，都会对计算结果带来致命影响。
2. 计算由一个或某几个结点承担。所有结点把加密后的数据发给一个结点，由这个结点进行所有的计算。优点是不需要考虑传递方案，缺点就是要求这个结点的可信度较高而且有较强的计算通信能力。

在我们的协议中，使用的是第二种方案，原因在于该方案比较简单易行，而且我们拥有计算通信能力较强的结点。值得注意的是，这个进行计算的结点不能是产生密钥的结点，否则该结点就能直接对其它结点发送来的数据进行解密得到原始数据，导致泄露其它结点隐私信息的泄露。

4.2 模型及参数说明

4.2.1 参数说明

ε：全同态加密机制。

ε_R：公钥加密机制，如 RSA 等。

n：感知层结点数。

N_i：第 i 个结点。

p_i：结点N_i的敏感数据，即明文。

ψ_i：p_i在ε中的密文。

(pk,sk)：ε的公钥与私钥。

(pk_R, sk_R)：ε_R的公钥与私钥。

4.2.2 加密方案

在我们的协议中，使用以下两个加密机制：

- 全同态加密机制ε，具体见第 2 章，由以下几个算法组成：

  $\boldsymbol{KeyGen}(\lambda)$：密钥产生算法。输入为秘密参数$\lambda$，输出为密钥对$(pk, sk)$。

  $\boldsymbol{Encrypt}(pk, p_i)$：加密算法。输入为公钥 pk 和明文p_i，输出为密文ψ_i。

  $\boldsymbol{Evaluate}(pk, f, (\psi_1, \dots, \psi_n))$：同态计算算法。输入为公钥 pk，函数（电路）f，密文集$(\psi_1, \dots, \psi_n)$，输出为$f(p_1, \dots, p_n)$在pk下的加密ψ。

  $\boldsymbol{Decrypt}(sk, \psi)$：解密算法。输入为私钥 sk 和密文ψ。输出为明文 p。

- 公钥加密机制ε_R，这里使用的是 ElGamal 加密方案，由以下几个算法组成：

  $\boldsymbol{KeyGen}_R(\lambda_R)$：密钥产生算法。输入为秘密参数$\lambda_R$，输出为密钥对$(pk_R, sk_R)$。算法根据$\lambda_R$产生一个阶为 q 生成元为 g 的循环群 G。从$\{0, \dots, q-1\}$中取一个随机数 x，设$h = g^x$，则公钥$pk_R = (g, h, q, G)$，私钥$sk_R = x$。

  $\boldsymbol{Encrypt}_R(pk_R, p_i)$：加密算法。输入为公钥$pk_R$和明文$\psi_i$，输出为密文$c_i$。随机从$\{0, \dots, q-1\}$中取一个数 y，密文$c_i = (c_{i1}, c_{i2}) = (g^y, p_i \cdot h^y)$。

  $\boldsymbol{Decrypt}_R(sk_R, c_i)$：解密算法。输入为私钥$sk_R$和密文$c_i$。输出为明文$p_i$。$p_i = c_{i2} \cdot (c_{i1}^x)^{-1} = p_i \cdot h^y \cdot (g^{xy})^{-1}$。

4.2.3 感知层结点分类

在我们的协议中，物联网感知层有以下三类不同的结点：

- 接收计算结果的结点为 V 结点，设为第 n-1 个结点，该结点负责产生ε的密钥对(pk, sk)，广播公钥pk，使用私钥sk把接收到的计算结果解密得到最终结果。

- 承担计算任务的结点为 R 结点，设为第 n 个结点，该结点负责接收其它结点的加密数据，并利用同态加密机制对这些数据进行计算，得到一个加密过的计算结果，并把它发给 V 结点。
- 除上述两个结点外，都为普通结点，设为$(N_1, N_2, \dots, N_{n-2})$。普通结点使用公钥加密自己的隐私数据，并发送给 R 结点进行计算。

4.2.4 计算结果接收者分类

在我们协议中，最终计算结果接收者有三类：感知层中的一个结点、感知层中的多个结点和上层。对不同的接收者有如下不同的处理：

- 如果是感知层中的一个结点，则这个结点为 V 结点。
- 如果是感知层中的 k 个结点，则从这 k 个结点中随机抽取一个结点作为 V 结点，协议结束时，由 V 结点把最终计算结果传给其它 k-1 个结点。
- 如果是上层，则从感知层中随机抽取一个结点作为 V 结点，协议结束时，由 V 结点把最终计算结果传给上层。

4.3 具体协议

我们的协议如下：

1. 结点 R 调用$\boldsymbol{KeyGen}_R(\lambda_R)$产生密钥对$(pk_R, sk_R)$，广播$pk_R$。
2. 结点 V 调用$\boldsymbol{KeyGen}(\lambda)$产生密钥对$(pk, sk)$，广播$pk$。
3. 结点$N_i, i = 1, \dots, n-2$，调用$\boldsymbol{Encrypt}(pk, p_i)$得到$\psi_i$，再调用$\boldsymbol{Encrypt}_R(pk_R, p_i)$得到$c_i$，并把$c_i$发送给 R 结点。结点 V 调用$\boldsymbol{Encrypt}(pk, p_{n-1})$得到$\psi_{n-1}$，把$\psi_{n-1}$发送给 R 结点。
4. 结点 R 收到$c_i, i = 1, \dots, n-2$，以及ψ_{n-1}后，调用$\boldsymbol{Decrypt}_R(c_i, sk_R)$得到$\psi_i$，并调用$\boldsymbol{Encrypt}(pk, p_n)$得到$\psi_n$，再调用$\boldsymbol{Evaluate}(pk, f, (\psi_1, \dots, \psi_n))$，得到$f(p_1, \dots, p_n)$在$pk$下的加密ψ，并把ψ发送给结点 V。
5. 结点 V 调用$\boldsymbol{Decrypt}(\psi, sk)$解密得到$f(p_1, \dots, p_n)$。

协议示意图如下：

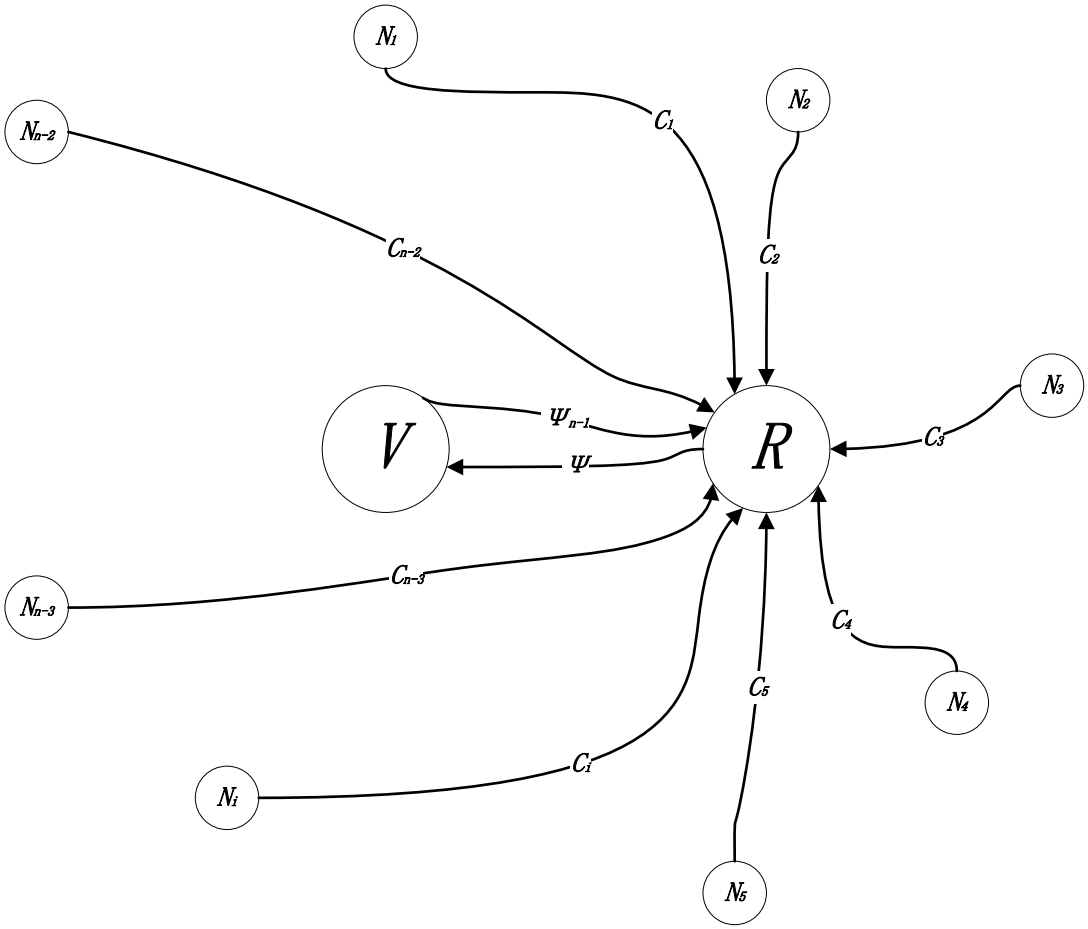

图4.1　基于安全多方计算的物联网感知层协议示意图

4.4 性能

在第二章，已经说明了 Gentry 全同态加密算法具有语义安全。而在 DDH 假设(decisional Diffie-Hellman assumption)下，ElGamal 是语义安全的。所以上述协议具有语义安全性。

而对于效率而言，ElGamal 的密文空间大小是明文空间大小的两倍，通信复杂度很高，可以考虑不加密整个用 Gentry 算法加密后的密文，而加密一个小的秘密参数来减小 ElGamal 算法中明文的大小，同时达到隐私性的要求。

4.5 改进

改进后的协议如下：

1. 结点 R 调用$\boldsymbol{KeyGen}_R(\lambda_R)$产生密钥对$(pk_R, sk_R)$，广播$pk_R$。
2. 结点 V 调用$\boldsymbol{KeyGen}(\lambda)$产生密钥对$(pk, sk)$，广播$pk$。
3. 结点$N_i, i = 1, \dots, n-2$，调用$\boldsymbol{Encrypt}(pk, p_i)$得到$\psi_i$，并随机生成一个小的参数$d_i$，调用$\boldsymbol{Encrypt}_R(pk_R, d_i)$得到$c_i$，并把$(\psi_i + d_i, c_i)$发送给 R 结点。结点 V 调用$\boldsymbol{Encrypt}(pk, p_{n-1})$得到$\psi_{n-1}$，把$\psi_{n-1}$发送给 R 结点。
4. 结点 R 收到$(\psi_i + d_i, c_i), i = 1, \dots, n-2$，以及$\psi_{n-1}$后，调用$\boldsymbol{Decrypt}_{\boldsymbol{R}}(c_i, sk_R)$得到$d_i$，从而可以得到$\psi_i$，接着调用$\boldsymbol{Encrypt}(pk, p_n)$得到$\psi_n$。这样就可以调用$\boldsymbol{Evaluate}(pk, f, (\psi_1, \dots, \psi_n))$，得到$f(p_1, \dots, p_n)$在$pk$下的加密ψ。最后把ψ发送给结点 V。
5. 结点 V 调用$\boldsymbol{Decrypt}(\psi, sk)$解密得到$f(p_1, \dots, p_n)$。

改进后的协议示意图如下：

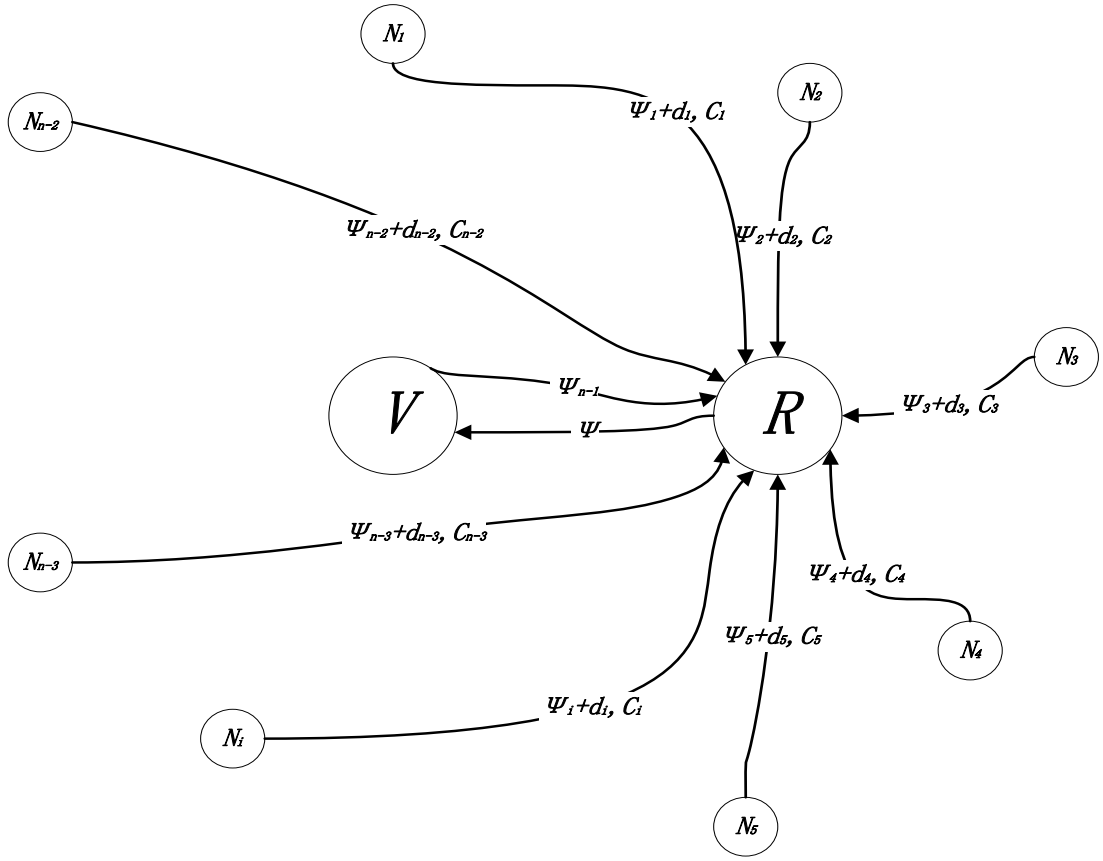

图4.2　基于安全多方计算的物联网感知层的改进协议示意图

4.6 协议分析

在改进后的协议中，使用 ElGamal 方案加密的明文由原来的计算数据变为了一个小的随机数，减小的密文大小也减小了计算复杂度。

我们的协议是基于 R 结点和 V 结点之间没有暗自串通的假设的，如果这一假设不成立，那么我们的协议就会泄露其它结点的信息。这一假设可以进一步弱化为 R 结点和 V 结点中至少有一个是可信任结点。

在我们的协议中，同态机制的密钥对(pk, sk)是由 V 结点产生的，生成密钥的时间不小。在每次计算结果的接收结点都不同的情况下，我们的协议就会接近于一次一密的情况，每次都要生成新的密钥，这对算法的效率有很大的影响。

R 结点和 V 结点是计算任务的主要承担结点，需要较强的计算和通信能力。在物联网感知层结点中，这样的结点可以设置多个，这样就可以尽可能地覆盖计算结果接收结点，减小生成重新生成密钥的时间。

我们的协议是基于密码学模型的，该模型比较符合物联网感知层的基本情况。但是我们的协议并没有考虑攻击模型。如果要考虑攻击模型，可以考虑使用私密分享的思想来实现。

4.7 本章小结

本章提出了基于安全多方计算的物联网隐私保护协议。本章先介绍了构造这样一个协议需要考虑的三个问题：结点能力、密钥生成、计算承担，并一一提出了解决方案。接下来介绍了协议所涉及到的参数，以及协议的模型，包括使用到的加密方案、结点的分类以及计算结果接收者的分类，给出了物联网感知层的基本构架。然后提出了基于上述构架的协议，并分析了该协议的性能以及缺点所在，根据该缺点提出了改进方案。最后分析了改进后的协议依旧存在的问题、整个模型中存在的问题以及解决方案。

第5章 总结与展望

5.1 研究工作总结

物联网是未来网络发展的趋势，它使人与人、物与物以及人与物之间的连接不再是梦想。在如此广泛的应用前景下，物联网隐私得到保证是物联网能广泛应用的基本要求之一。目前对物联网隐私工作的研究还出于刚起步的状态，研究成果较少。

本论文的主要工作是对物联网感知层的隐私保护机制进行了研究与探索。本论文的主要研究内容与成果有：

1. 归纳了物联网及其在隐私保护要求下的相关背景与问题。介绍了物联网的特点以及安全性要求，物联网感知层的能力对安全实现的限制与瓶颈，以及可行的解决方案——利用基于全同态加密的安全多方计算，以达到物联网感知层的隐私保护目的。
2. 回顾了已有的全同态加密算法，分析了它们的优点与缺点。深入研究了 Gentry 全同态加密算法。该算法能对任意电路进行有效计算，而且不存在密文规模指数爆炸的问题，适用物联网感知层结点通信计算能力有效的情况。其关键思想主要有三点：首先是电路的可自启性，该性质保证了算法能处理任意函数；其次是理想格的利用，理想格为算法的可自启性以及同态性提供了保证；最后是压缩解密电路，通过解密电路的压缩，提高了可处理电理的深度，减小了算法复杂度。
3. 总结了安全多方计算的定义、攻击者模弄以及通信模型，分析了现有安全多方计算方案的优点与缺点。
4. 提出了基于安全多方计算的物联网感知层安全模型。从物联网感知层安全模型的基本要求入手，介绍了协议需要考虑的三个问题：结点能力、密钥生成、计算承担，并一一提出了解决方案。接下来介绍了协议所涉及到的参数，以及协议的模型，包括使用到的加密方案、结点的分类以及计算结果接收者的分类，给出了物联网感知层的基本构架。然后提出了基于上述构架的协议，并分析了该协议的性能以及缺点所在，根据该缺点提出了改

进方案。最后分析了改进后的协议依旧存在的问题、整个模型中存在的问题以及解决方案。

5.2 研究展望

物联网的隐私保护研究才刚刚起步，需要研究的内容还有很多。在以后的工作中，我将从以下几个方面进行研究：

1. 实用的物联网隐私保护机制。将已有的算法通过仿真模拟实际的系统，进行可行性分析。
2. 协议的性能改进。
3. 考虑不同的攻击模型，对不同攻击模型进行协议改进，或者提出新的协议和算法。

插图索引

参考文献

[1] 丁超，杨立君，吴蒙. IoT-CPS 的安全体系结构及关键技术[J]. 中兴通讯技术, 2011,17(1):11-16.

[2] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems[J].Communications. of the ACM, 1978,21(2): 120–126.

[3] R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphisms[M]. Foundations of Secure Computation, 1978: 169–180.

[4] E. Brickell and Y. Yacobi. On Privacy Homomorphisms[C]//Proceedings of Eurocrypt ’87. LNCS 304, Springer, 1988: 117–125.

[5] D. Boneh and R. Lipton. Searching for Elements in Black-Box Fields and Applications[C]//Proceedings of Crypto ’96. LNCS 1109, Springer, 1996: 283–297.

[6] S. Goldwasser and S. Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information[C]//Proceedings of the 14th Annual ACM Symp. on the Theory of Computing (STOC), ACM.1982: 365–377.

[7] J. Benaloh. Verifiable secret-ballot elections [D]. Yale University, Dept. of Computer Science, New Haven, CT, USA. 1988.

[8] D. Naccache and J. Stern. A New Public-Key Cryptosystem Based on Higher Residues[C] // ACM Conference on Computer and Communications Security (CCS)’98. 1998.

[9] T. Okamoto and Uchiyama. A New Public-Key Cryptosystem as Secure as Factoring[C] // Proceedings of Eurocrypt ’98. LNCS 1403, Springer, 1998: 308–318.

[10] P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes[C] // Proceedings of Eurocrypt ’99, 1999:223–238.

[11] T. ElGamal. A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms[C] // Proceedings of Crypto ’84, 1984: 469–472.

[12] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts[C] // Theory of Cryptography Conference (TCC) 2005. LNCS 3378, Springer, 2005: 325–341.

[13] M. Fellows and N. Koblitz. Combinatorial cryptosystems galore [M]. Contemporary Mathematics, vol. 168 of Finite Fields: Theory, Applications, and Algorithms, FQ2, 1993: 51–61.

[14] T. Sander, A. Young, and M. Yung. Non-interactive crypto computing for NC1[C] // Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS), 1999: 554–567.

[15] L. Ly. A public-key cryptosystem based on Polly Cracker[D]. Ruhr University Bochum, Bochum, Germany 2002.

[16] C. Gentry. Fully homomorphic encryption using ideal lattices[C] // Proceedings of the 41st annual ACM symposium on Theory of computing (STOC), 2009.

[17] S. Goldwasser. Multi party computations: past and present[C] // Proceedings of the 16th annual ACM symposium on Principles of distributed computing. 1997:1-6.

[18] R. Cramer and I. Damgard. Multiparty Computation, an Introduction[M]. Contemporary Cryptography. Birkhuser Basel, 2005:7-8.

[19] A. Yao. Protocols of secure computations [C] // Proceedings of the 23rd IEEE Symp. On the Foundation of Computer Science (FOCS). IEEE, 1982: 160-164.

[20] M. Ben-Or, S. Goldwasser, A. Wigderson: Completeness theorems for Non-Cryptographic Fault-Tolerant Distributed Computation[C] // Proceedings of 20th Annual ACM Symp. on the Theory of Computing (STOC)'88. 1988: 1–10.

[21] D. Chaum, C. Crépeau, I. Damgård: Multi-Party Unconditionally Secure Protocols[C] // Proceedings of 20th Annual ACM Symp. on the Theory of Computing (STOC)'88. 1988:11–19.

[22] T. Rabin, M. Ben-Or: Verifiable Secret Sharing and Multiparty Protocols with Honest majority[C] // Proceedings of 20th Annual ACM Symp. on the Theory of Computing (STOC) '89. 1989: 73–85.

[23] I. Damgård, J.B. Nielsen. Scalable and Unconditionally Secure Multiparty Computation[C] // Proceedings of Crypto'07. 2007: 572–590.

[24] M. Hirt, U. Maurer. Complete Characterization of Adversaries Tolerable in General Multiparty Computations[C] // Proceedings of 16th Annual ACM Symp. on Principles of Distributed Computing (PODC)'97. 1997: 25–34.

[25] R. Canetti, Y. Lindell, R. Ostrovsky, A. Sahai. Universally composable two-party and multi-party secure computation[C] // Proceedings of 33th Annual ACM Symp. on the Theory of Computing (STOC) '02. 2002, pp. 494-503.

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____　　日　期：_____

附录 A 外文资料的调研阅读报告

Exploration of Privacy-Preserving Scheme in Internet of Things

The Internet Today is moving beyond just connecting millions of computers and web sites towards connecting billions of countless physical objects. From e-commerce, to e-everything, the Internet is moving into the future Internet of smart vehicles, smart phones, smart offices, smart homes, smart schools, smart factories and smart government to smart everything. With computational power and digital communications embedded in every object around human beings, a new kind of Internet, the Internet of Things, is being created, which makes new kind of ubiquitous applications possible. They will come into people's everyday life and make great improvement to the security and the quality of human beings' lives.

To ensure that the future Internet, the Internet of Things, is built on basis that will not easily collapse, it's very important to prepare for the possible threat scenarios, which come with the emerging and future developments, and construct some efficient schemes to handle these scenarios.[5]

The Internet of Things is so critical that it cannot be built on bolted existing security solutions. It must be built on the top of a very strong basis with security as one of its most important priorities. This must be achieved in a proactive way, by anticipating the threats which may happen in the future. And then develop possible countermeasures to handle them.

Undoubtedly, Internet of Things will create tremendous applications and opportunities to improve peoples' everyday lives. Most of these ubiquitous applications need to be performed with inputs provided by different untrusted parties. However, since these inputs contain private information about people's daily life, how to perform these computations securely and correctly becomes a critical problem. With the same background, secure multi-party computations may be an efficient way to solve this problem and to protect peoples' privacy in Internet of Things.

Secure multi party computation is a sub field of cryptography. The goal of methods for secure multi party computation is to enable parties to jointly compute a function over their inputs, while at the same time keeping these inputs private[2]. For example, two millionaires can compute which one is richer, but without revealing

their net worth. In fact, this very example was initially suggested by Andrew C. Yao[4] in a 1982 paper and was later named the millionaire problem.Concretely, assuming we have inputs $x_1, \dots, x_n$, where party *i* knows x_i, and we want to compute $f(x_1, \dots, x_n) = (y_1, \dots, y_n)$ such that party *i* is guaranteed to learn y_i, but can get nothing more than y_i.

It is obvious that if we can compute any function securely, we have a very powerful protocol. However, some problems require more general ways to solve them. For example, a secure payment system cannot naturally be formulated as secure multi party computation of merely a single function. We need to keep track of the money each party owns and prevent that some parties spend much more money than they have. This payment system should work as a secure general-purpose system, which can receive inputs from different parties at different time and produce results for each existing parties with the current inputs and previously stored data. Therefore, the solutions of secure multi party computation protocols need to be more general. Homomorphic encryption can be used as a basic building block in secure multi party computation protocols.

Homomorphic encryption is a specific form of encryption where a specific algebraic operation performed on the plaintext is equivalent to another algebraic operation performed on the ciphertext. This property can be seen as either a positive or negative attribute of the cryptosystem.

RSA encryption was the first partially homomorphic encryption with multiplicatively homomorphism. However, it is not semantically secure. Elgamal encryption is another multiplicative homomorphic encryption. Homorphic encryptions with additively homomorphism are Paillier encryption, Benaloh encryption, Goldwasser-Micali encryption and etc.

Each of the homomorphic encryption listed above allows homomorphic computation of only one operation (either addition or multiplication) on plaintexts. A cryptosystem which supports both addition and multiplication is known as fully homomorphic encryption and is far more powerful.

In 2009, Craig Gentry [1]constructs the first fully homomorphic encryption scheme based on lattice-based cryptography. His scheme allows evaluations of arbitrary depth circuits. In other words, the scheme supports any functions computed on ciphertext. Gentry's scheme starts with a somewhat homomorphic encryption scheme using ideal lattices, which can only evaluate low depth circuits. Then he

shows how to make the original scheme bootstrappable, which means that the scheme can evaluate arbitrary circuit with a fixed depth. Firstly, he makes a slight modification to the original scheme so that it can evaluate its own decryption circuit, which is a self-referential property of the scheme. Then, he proves that any bootstrappable somewhat homomorphic encryption scheme can be modified into a fully homomorphic encryption by a recursive self-embedding. The security of the fully homomorphic encryption scheme is based on the assumed hardness of two problems: certain worst-case problems over ideal lattices and the sparse subset sum problem.

Considering performance of Gentry's scheme, ciphertexts' size will not explode with increase of the depth of the evaluated circuit. That is to say, the ciphertext is independent of the complexity of the function, which is evaluated over the encrypted data. However, the scheme is impractical for many applications, because ciphertext size along with computation time will increase sharply, when increasing the security level.

References

[1] C. Gentry. Fully homomorphic encryption using ideal lattices[C] // Proceedings of the 41st annual ACM symposium on Theory of computing (STOC), 2009.

[2] S. Goldwasser. Multi party computations: past and present[C] // Proceedings of the 16th annual ACM symposium on Principles of distributed computing. 1997:1-6.

[3] R. Cramer and I. Damgard. Multiparty Computation, an Introduction[M]. Contemporary Cryptography. Birkhuser Basel, 2005:7-8.

[4] A. Yao. Protocols of secure computations [C] // Proceedings of the 23rd IEEE Symp. On the Foundation of Computer Science (FOCS). IEEE, 1982: 160-164.

[5] Oleshchuk V. Internet of Things and Privacy Preserving Technologies [C]//Proceedings of the 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronics Systems Technology(Wireless VITAE'09). IEEE, 2009: 336-340.

综合论文训练记录表

论文题目	
主要内容以及进度安排	指导教师签字： 考核组组长签字： 年　　月　　日
中期考核意见	考核组组长签字： 年　　月　　日

指导教师评语	指导教师签字：_____ 年　　月　　日
评阅教师评语	评阅教师签字：_____ 年　　月　　日
答辩小组评语	答辩小组组长签字：_____ 年　　月　　日

总成绩：_____

教学负责人签字：_____

年　　月　　日