a*b具体分析

- ABY phase总共分了

  P_INIT, P_CIRCUIT, P_NETWORK, P_BASE_OT, P_OT_EXT, P_GARBLE, P_ONLINE 7个阶段

- new ABYPatty // new party
  - 指针初始化m_cCrypt(std::make_unique(seclvl.symbits)), //一个密码学指针 glock(std::make_unique()), // 时间？？ m_eMTGenAlg(mg_algo), //mts方式

    m_eRole(pid), //角色 m_nNumOTThreads(nthreads),//ot线程

    m_tComm(std::make_unique<comm_ctx>()), //comm有4个指针

    m_pSetup(std::make_unique(m_cCrypt.get(),

    m_nNumOTThreads, m_eRole, m_eMTGenAlg)), m_nPort(port), //端口

    m_sSecLvl(seclvl),//安全等级 m_cAddress(addr)
  - P_INIT //没有参数，返回一个bool
    - Init(); //m_vSockets = 2 ，建立2个连接一个receiver一个sender 初始化了 m_nMyNumInBits = 0; m_nHelperThreads = 2; m_vSockets.resize(2); 返回一个true

      m_pCircuit = NULL;

    P_INIT STOP
  - P_CIRCUIT //初始话算术电路参数
    - InitCircuit(bitlen, reservegates, abycircdir) //bitlen为位数，reservegates为最大电路门个数，abycircdir 为一个电路路径 输出为../../bin/circ/ 输出一个bool
      - m_pCircuit = new ABYCircuit(reservegates); //m_nMaxVectorSize{1}, m_nMaxDepth{0}

        这是初始化电路参数s_bool，S_YAO, S_ARITH, 算术以bitlen位数确定生成电路以32为例
      - m_vSharings[S_ARITH] = new ArithSharing<uint32_t>(S_ARITH, m_eRole, 1, m_pCircuit, m_cCrypt.get(), m_eMTGenAlg); // S_ARITH代表电路 ，m_eRole角色，sharebitlen，m_pCircuit，m_cCrypt.get()为密码参数，m_eMTGenAlg为mts构造方式
        - Init（）初始化算术电路参数
        - new ArithmeticCircuit 里面继续Init（）这个初始话是and门，cons门的个数

    P_CIRCUIT STOP
- ABYPatty STOP
- party->GetSharings(); //获取输入电路类型
- (ArithmeticCircuit*) sharings[S_ARITH] //一个算术电路
- Arith_circ->PutINGate //输入一个值
- Arith_circ->PutMULGate //计算乘法
- ABYParty::ExecCircuit()
  - ConnectAndBaseOTs();
    - P_NETWORK

        - EstablishConnection（）//建立网络连接，m_tComm
    - P_NETWORK STOP
    - P_BASE_OT
        - PrepareSetupPhase（） //传入参数 m_tComm指针

            m_tSetupChan = new channel 建立了一个setup channel通道

            初始化BaseOT参数

            若使用MT_PAILLIER时该阶段生成密钥m_cPaillierMTGen->keyExchange(m_tSetupChan
    - P_BASE_OT STOP
- P_TOTAL
    - P_SETUP

      m_vSharings[i]->PrepareSetupPhase(m_pSetup.get()）//遍历执行各个电路的PrepareSetup

      主要以Arith的PrepareSetupPhase（m_pSetup.get）为例

      m_nMTS 乘法三元组个数

      InitMTs()//用随机值填充m_vA ,m_vB ； m_vC,m_vS填充了m_nMTs*m_nTypeBitLen个0

      其他初始化大小为1

      然后搞了个结构体指针 pgentask = (PKMTGenVals*) malloc(sizeof(PKMTGenVals));

      将A，B，C，m_nMTs, m_nTypeBitLen赋值进去，然后push_back到vector<PKMTGenVals*> m_vPKMTGenTasks;
        - P_OT_EXT

          m_pSetup->PerformSetupPhase(); //此阶段 主要是OT_EXT和在线计算乘法三元组

            WakeupWorkerThreads(e_MTPaillier);

                switch (job)

                    bSuccess = m_pCallback->ThreadRunPaillierMTGen(threadid);

                        初始化参数，线程参数等

                            根据角色computeArithmeticMTs

            success &= WaitWorkerThreads(); //
        - P_OT_EXT STOP
        - P_GARBLE

          [S_YAO]->PerformSetupPhase/这个阶段是初始化yao 跳过
        - P_GARBLE STOP
    - P_SETUP STOP
    - P_ONLINE

      EvaluateCircuit() //在线计算

      m_tPartyChan = new channel

m_vSharings[i]->PrepareOnlinePhase();//遍历电路 每个电路都跑 主要看一下算术电路

//获取自己输入总bitlen，输出总bitlen ，获取另外一方

//初始化m_vInputShareSndBuf 随机数

//m_vOutputShareSndBuf， m_vInputShareRcvBuf，m_vOutputShareRcvBuf 为0

InitNewLayer()

// m_vInputShareSndBuf //掩码

//其他部分 Evaluate Circuit layerwise

- P_ONLINE STOP

- P_TOTAL