**Customer Process**

```haskell
let CusProcess =
let Pc = Pmulti(Dc, Q) in
out(PubChCB, (Pc));
out(PubChCM, (Pc));
in(PubChCB, (XPb: bitstring));
in(PubChCM, (XPm: bitstring));
new r: bitstring;
new T1: bitstring;
event pstartCustomer(IDc);
let R = Pmulti(Padd(Dc, T1), Minver(r)) in
let Kx = Pmulti(r, XPb) in
let m = Owhf(stconcat(GI, stconcat(p, IDb))) in
let C1 = Sencr((IDc, m, Xcord(Kx), T1), Xcord(Kx)) in
out(PubChCB, (C1, R, T1)); (*To bank*)
in(PubChCB, (XC2: bitstring, XT2: bitstring));
let (XDs: bitstring, XE: bitstring, XXkx: bitstring, XXT2: bitstring) = symd(XC2, Xcord(Kx)) in
if (XT2=XXT2)
then if (Xcord(Kx)=XXkx)
then new r1: bitstring;
new T3: bitstring;
let R1 = Pmulti(Padd(Dc, T3), Minver(r1)) in
let Kx1 = Pmulti(r1, XPm) in
let C3 = Sencr((IDb, XDs, XE, GI, Xcord(Kx1), T3), Xcord(Kx1)) in
out(PubChCM, (C3, R1, T3)); (*To merchant*)
event pendCustomer(IDc).
```

**Bank**

```haskell
let BanProcess =
let Pb = Pmulti(Db, Q) in
in(PubChCB, (XPc: bitstring));
out(PubChCB, (Pb));
in(PubChCB, (XC1: bitstring, XR: bitstring, XT1: bitstring));
event pstartBank(IDb);
let Kx = Pmulti(Padd(Pmulti(Db, XPc), Pmulti(Pb, XT1)), XR) in
let (= IDb, XDs: bitstring, XE: bitstring, GI, Xkx: bitstring, XXT1: bitstring) = symd(XC1, Xcord(Kx)) in
if (Xcord(Kx)=Xkx) then if (XT1=XXT1)
then new E: bitstring;
new T2: bitstring;
let M = stconcat(Xm, E) in
let DS = Ssig(M) in
let C2 = Sencr((DS, E, Xkx, T2), Xkx) in
out(PubChCB, (C2, T2));
event pendBank(IDb).
```

**Merchant**

```haskell
let MerProcess =
in(PubChCM, (XPc: bitstring));
let Pm = Pmulti(Dm, Q) in
out(PubChCM, (Pm));
in(PubChCM, (XC3: bitstring, XR1: bitstring, XT3: bitstring));
event pstartMerchant(IDm);
let Kx1 = Pmulti(Padd(Pmulti(Dm, XPc), Pmulti(Pm, XT3)), XR1) in
let (= IDb, XDs: bitstring, XE: bitstring, GI: bitstring, Xkx: bitstring, XXT3: bitstring) = symd(XC3, Xcord(Kx1)) in
if (XT3=XXT3) then if (Xcord(Kx1)=Xkx)
then let m = Owhf(stconcat(GI, stconcat(p, IDb))) in
let M = stconcat(m, XE) in
if (Ssig(M)=XDs) then let C4 = Sencr(GI, Xcord(Kx1)) in
out(PubChCM, (C4));
event pendMerchant(IDm).
```

Fig. 9. Process part