# 门限多方 ECDSA 方案

## . 密钥生成阶段

此阶段主要生成门限多方 ECDSA 所需要的每个参与方的公私钥对、签名公钥、以及每个用户的签名私钥的分享（share）。我们的方案是一个 (t,n)-门限方案，一共有 $n$ 个参与方，当有超过 $t$ 个参与方同意（不包含 $t$），则可以生成一个有效签名。以下是具体过程：

第一步：每一个参与方 $P_i$ ($i \in [n]$)，和两方方案的做法一样，生成一对 promise 加密的公私钥对（包括 Elgamal 的公私钥和 CL-Elgamal 的公私钥），我们记为 $(\mathrm{pk}_i, \mathrm{sk}_i)$。

第二步：每一个 $P_i$，随机的选择 $u_i \leftarrow Z/qZ$ 并且计算承诺 $\mathrm{kgc}_i \leftarrow \mathrm{Com}(u_i P)$。然后 $P_i$ 将 $(\mathrm{pk}_i, \mathrm{kgc}_i)$ 广播出去。

第三步：每一个 $P_i$，收到其他所有参与方发送的公钥和承诺后，令 $Q_i = u_i P$，将 $\mathrm{kgc}_i$ 打开并广播出去。$P_i$ 收到来自于其他所有参与方的承诺打开信息后，验证承诺是否正确，如果存在一个承诺验证不通过则终止协议，否则，计算 $Q = \sum_{i \in [n]} Q_i$。$Q$ 即是签名公钥。

第四步：每一个 $P_i$，随机生成一个 $t$ 次的多项式 $p_i(X) = u_i + \sigma_{k \in [t]} a_{i,k} X^k \bmod q$（通过随机的选取每一项的系数 $a_{i,k}$，然后计算 $\sigma_{i,j} = p_i(j)$ 对于所有的 $j \in [n]$，然后计算 $V_{i,k} = a_{i,k} P$ 对于所有的 $k \in [t]$。$P_i$ 将 $\{V_{i,k}\}_{k \in [t]}$ 广播出去，并且通过两方信道将 $\sigma_{i,j}$ 发送给参与方 $P_j$。

第五步：每一个 $P_i$:

- 对于所有的 $j \neq i, j \in [n]$，验证 $\sigma_{j,i}$ 是否等于 $Q_j \prod_{k \in [t]} V_{j,k}^{i^k}$。如果存在一个 $j$ 使得验证不通过，则终止协议。
- 计算 $x_i = \sum_{k \in [n]} \sigma_{k,i}$。
- 计算 $X_i = x_i P$, 并且生成一个与之相关的 NIZK 证明 $\pi_{kg,i}$。$\pi_{kg,i}$ 证明的断言是 “存在一个 $x_i$，使得 $X_i = x_i P$。”（这是为了证明 $X_i$ 是正确生成的。）

- 将 $X_i$ 和 $\pi_{kg,i}$ 广播出去。

第六步：每一个 $P_i$，收到所有的其他参与方从第五步传来的消息后，对于所有的 $j \in [n], j \neq i$，验证 $X_j$ 和 $\pi_{kg,j}$ 是否正确，如果存在一个验证通不过，则终止协议。

. 签名阶段 在此阶段我们用密钥生成阶段生成的信息对消息 $m$ 进行签名，我们记 $S \subset [n]$ 为参与这次签名过程的参与方的集合。具体过程如下：

预处理：对于 $i \in S$, 每一个 $P_i$:

- 计算 $\lambda_i = \prod_{j \in S, j \neq i}(-j) / \prod_{j \in S, j \neq i}(i-j)$。
- 计算 $W_k = \lambda_k X_k$ 对于所有的 $k \in S$, 且计算 $w_i = \lambda_i x_i$。（所有的 $\{X_k\}_{k \in S}$ 都是公开的，而 $P_i$ 只知道自己的 $x_i$, 所以 $P_i$ 可以计算 $\{W_k\}_{k \in S}$ 和 $w_i$。）

经过预处理后会有这样的关系 $\sum_{k \in S} W_k = Q$ 成立。

第一步：对于 $i \in S$，$P_i$:

- 随机选择 $k_i, \gamma_i \leftarrow Z_q$ 以及 $r_i \leftarrow [A]$($r_i$ 是 promise 加密所用到的随机数，所以需要从一个大的整数范围里选取，和两方方案一样。)
- 计算密文 $c_{k_i} \leftarrow \mathrm{Enc}(\mathrm{pk}_i, k_i; r_i)$，并且生成一个 NIZK 证明 $\pi_i$。$\pi_i$ 是用我们的 promise sigma 协议来生成的。然后计算承诺 $c_i \leftarrow \mathrm{Com}(\gamma_i P)$。
- 将 $(c_i, c_{k_i}, \pi_i)$ 广播出去。
- 校验所有收到的 $(c_{k_i}, \pi_i)$。

第二步：对于所有的 $i \in S$，$P_i$ 和其他每个 $P_j$ $(j \in S, j \neq i)$ 都要通过两方信道进行一次两方的交互:

- $P_i$ 随机的选取 $\beta_{j,i}, v_{j,i} \leftarrow Z_q$，并计算 $B_{j,i} = v_{j,i} P$；随机选择 $t_{j,i}, \hat{t}_{j,i} \leftarrow [A]$，计算 $t_{p,ji} = t_{j,i} \bmod q, \hat{t}_{p,ji} = \hat{t}_{j,i} \bmod q$，利用 promise 加密的同态性质和在第一步中得到的密文 $c_{k_j}$ 来计算密文 $c_{k_j \gamma_i}$ 和 $c_{k_j w_i}$，它们对应的明文分别应该是 $k_j(\gamma_i + \hat{t}_{j,i}) - \beta_{j,i}$ 和 $k_j(w_i + t_{j,i}) - v_{j,i}$。$P_i$ 将 $(c_{k_j \gamma_i}, c_{k_j w_i}, B_{j,i}, t_{p,ji}, \hat{t}_{p,ji})$ 发送给 $P_j$。
- $P_j$ 收到 $P_i$ 的消息后，对密文解密得到 $\alpha_{j,i1}$ 和 $\mu_{j,i1}$（其中 $\alpha_{j,i} = k_j(\gamma_i + \hat{t}_{j,i}) - \beta_{j,i} \bmod q$ 以及 $\mu_{j,i} = k_j(w_i + t_{j,i}) - v_{j,i} \bmod q$），从而 $P_j$ 可以利用自己知道的 $k_j, t_{p,ji}, \hat{t}_{p,ji}$ 来计算 $\alpha_{j,i} = k_j \gamma_i - \beta_{j,i} \bmod q$ 和 $\mu_{j,i} = k_j w_i - v_{j,i} \bmod q$。
- $P_j$ 验证 $k_j W_i$ 是否等于 $\mu_{j,i} P + B_{j,i}$，如果验证失败，则终止协议。

在所有的这些点对点的交互都完成后，对于 $i \in S$，每一个 $P_i$ 都计算 $\delta_i = k_i \gamma_i + \sum_{j \neq i}(\alpha_{i,j} + \beta_{j,i})$ 以及 $\sigma_i = k_i w_i + \sum_{j \neq i}(\mu_{i,j} + v_{j,i})$。

第三步：对于 $i \in S$，所有的 $P_i$ 将 $\delta_i$ 广播出去。在收到所有的广播消息后，计

算 $\delta = \sum_{i \in S} \delta_i$。

第四步：对于 $i \in S$，所有的 $P_i$:

- 将承诺 $c_i$ 的打开信息广播出去。(在第一步，是对 $\gamma_i P$ 的承诺)。
- 不妨记 $\Gamma_i = \gamma_i P$, 对断言 "存在一个 $\gamma_i$，使得 $\Gamma_i = \gamma_i P$" 生成一个 NIZK 证明 $\pi_{i\gamma}$。将 $\pi_{i\gamma}$ 广播出去。
- 对于收到的所有的 $\{c_j\}_{j \in S, j \neq i}$ 和 $\{\pi_{j\gamma}\}_{j \in S, j \neq i}$，验证证明是否正确，如果存在一个验证失败，则终止协议。
- 计算 $R = \delta^{-1}(\sum_{i \in S} \Gamma_i)$，然后令 $r = rx$，其中 $rx$ 是 $R$ 的横坐标（即 R 可以写成 $(rx, ry)$)。

第五步：对于 $i \in S$，每一个 $P_i$:

- 计算 $s_i = k_i m + \sigma_i r$, 随机选择 $\rho_i, l_i \leftarrow Z_q$, 并计算 $V_i = s_i R + l_i P$, $A_i = \rho_i P$; 以及承诺 $c1_i \leftarrow \mathrm{Com}(V_i, A_i)$。将 $c1_i$ 广播出去。
- 在接收到所有的承诺 $\{c1_j\}_{j \in S, j \neq i}$ 后，将 $c1_i$ 打开，然后对断言"存在 $(s_i, \rho_i, l_i)$, 使得 $(V_i = s_i R + l_i P) \wedge (A_i = \rho_i P)$" 生成一个 NIZK 证明 $\pi_{iVA}$。将 $c1_i$ 的打开信息和 $\pi_{iVA}$ 广播出去。
- 在收到所有的打开信息 $\{V_j, A_j\}_{j \in S, j \neq i}$ 和相应的 NIZK 证明 $\{\pi_{jVA}\}_{j \in S, j \neq i}$ 之后，验证这些承诺是否正确，以及左右的零知识证明是否正确，如果存在验证失败，则终止协议。
- 计算 $V = -mP - rQ + \sum_{i \in S} V_i$ 和 $A = \sum_{i \in S} A_i$。然后计算 $U_i = \rho_i V$, $T_i = l_i A$, $c2_i \leftarrow \mathrm{Com}(U_i, T_i)$。将 $c2_i$ 广播出去。
- 在收到所有的 $\{c2_j\}_{j \in S, j \neq i}$ 之后，将 $c2_i$ 打开并广播。
- 收到所有的打开信息后，验证所有的承诺是否正确，如果存在验证失败，则终止协议；否则，验证 $\sum_{i \in S} T_i$ 是否等于 $\sum_{i \in S} A_i$，验证失败则终止协议。
- 将 $s_i$ 广播出去。
- 在收到所有的 $\{s_j\}_{j \in S, j \neq i}$ 之后，计算 $s = \sum_{i \in S} s_i$。检查 $(r, s)$ 是不是 $m$ 的一个合法签名，如果是，则输出 $(r, s)$，否则，终止协议。