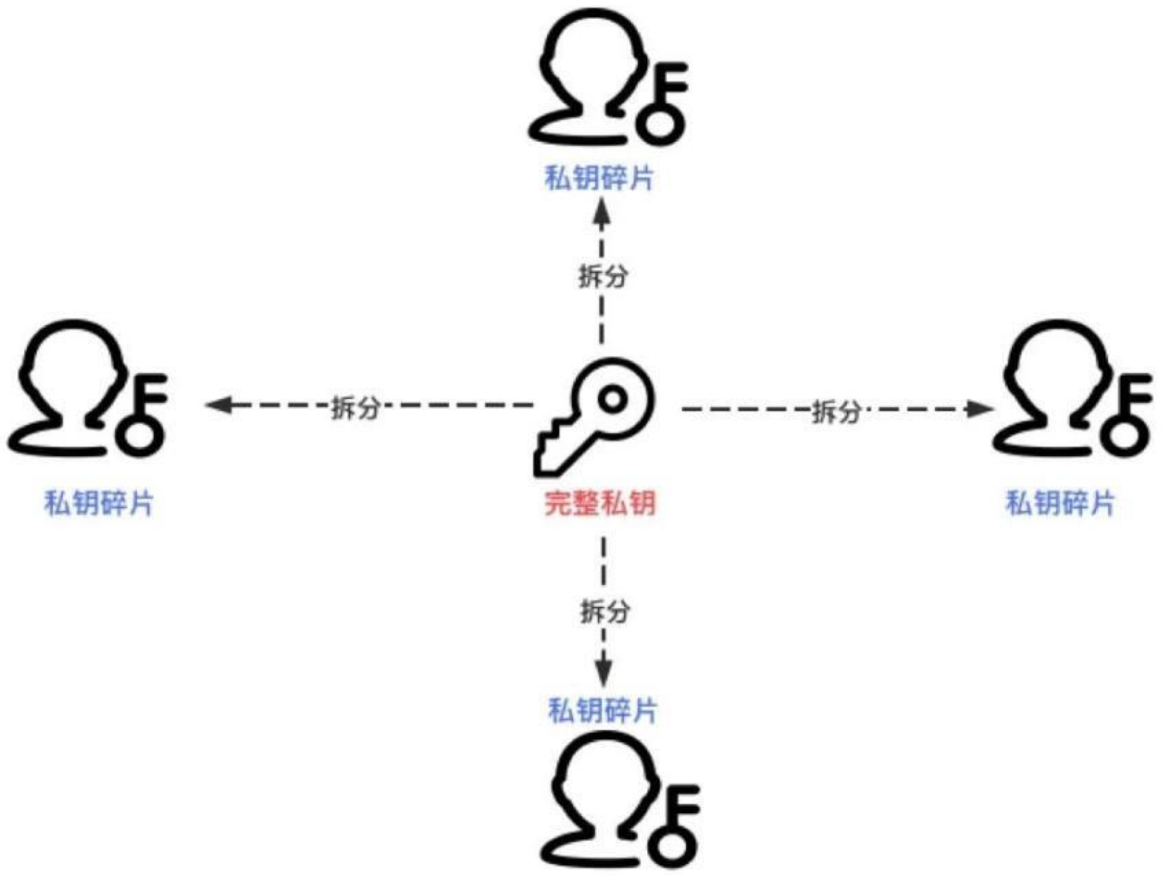
Threshold ECDSA in Three Rounds

Authors: Jack Doerner;
Yashvanth Kondi;
Eysa Lee;
Abhi Shelat

Published in: 2024 IEEE Symposium on Security and Privacy

◆ Introduction



◆ Introduction

ECDSASign(sk,m):

$r \leftarrow Z_q$

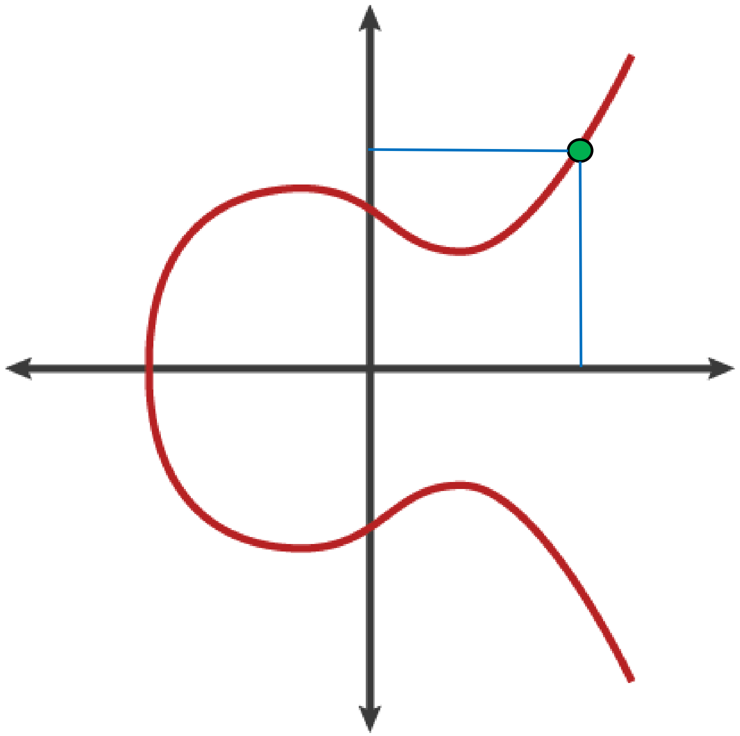
$R = r \cdot G$

$e = H(m)$

$s = \frac{e}{r} + \frac{sk \cdot r_x}{r}$

$\sigma = (s, R)$

output σ



Introduction

Threshold ECDSA difficulty：

ECDSASign(sk,m):

$$r \leftarrow Z_q$$

$$R = r \cdot G$$

$$e = H(m)$$

$$s = \frac{e}{r} + \frac{sk \cdot r_x}{r}$$

$$\sigma = (s, R)$$

output σ

模乘逆元	三个r一致
秘密值乘法	sk与公钥PK一致

不存在s的简单线性分解

Introduction

Inverted Nonce Rewriting

ECDSASign(sk,m):

$$[r] \leftarrow Z_q$$

$$R = r \cdot G \quad \rightarrow \quad R = [r^{-1}] \cdot G$$

$$e = H(m)$$

$$s = \frac{e}{r} + \frac{sk \cdot r_x}{r} \quad \rightarrow \quad s = (e + [sk] \cdot r_x)[r]$$

$$\sigma = (s, R)$$

output σ

Introduction

Inverted Nonce Rewriting

ECDSASign(sk,m):

$[r] \leftarrow Z_q$

$[\phi] \leftarrow Z_q$

reveal $[\phi] \cdot [r]$

reveal $\Phi = [\phi] \cdot G$

$R = (\phi r)^{-1} \cdot \Phi = [r^{-1}] \cdot G$

$e = H(m)$

$s = (e + [sk] \cdot r_x)[r] \quad \rightarrow \quad s = (\frac{a}{r}) + (\frac{b \cdot sk}{r})$

output (s, R)

Introduction

Advantage

Simplicity	commitments + multiplication(VOLE)
Security	threshold security + VOLE(OT) security
Efficiency	three rounds —pipelining→ two rounds

Introduction

Rewriting ECDSA

ECDSASign([sk],m):

$$[r] \leftarrow Z_q, [\phi] \leftarrow Z_q$$

$$R = \text{Reveal } [r] \cdot G$$

$$e = H(m)$$

$$s = \text{Reveal} \frac{e + sk \cdot r_x}{r} \cdot \frac{[\phi]}{[\phi]}$$

$$\sigma = (s, R)$$

output σ

◆ Introduction

Rewriting ECDSA

ECDSASign([sk],m):

$[r] \leftarrow Z_q, [\phi] \leftarrow Z_q$

$R = \text{Reveal } [r] \cdot G$

$e = H(m)$

u 和 s 决定

$w = \text{Reveal } e \cdot [\phi] + r_x \cdot [sk \cdot \phi]$

ϕ 掩盖 r

$u = \text{Reveal } [r \cdot \phi]$

两个k一致

$s = w/u$

$\sigma = (s, R)$

output σ

◆ Introduction

ECDSASign([sk],m):

$[r] \leftarrow Z_q, [\phi] \leftarrow Z_q$

$R = \text{Reveal } [r] \cdot G$

$e = H(m)$

$w = \text{Reveal } e \cdot [\phi] + r_x \cdot [sk \cdot \phi]$

$u = \text{Reveal } [r \cdot \phi]$

$s = w/u$

$\sigma = (s, R)$

output σ

Adversary	Defend
$[r] \leftarrow Z_q$	a commitment for R_i
$[sk \cdot \phi], [r \cdot \phi]$	a two-output VOLE
$[sk \cdot \phi], [r \cdot \phi]$	Verify Consistency by ϕ
$[sk \cdot \phi], [r \cdot \phi]$	check $s \cdot G = \frac{e+sk \cdot r_x}{r} \cdot \frac{[\phi]}{[\phi]} \cdot G = \frac{e \cdot G + Pk \cdot r_x}{R} \cdot \frac{[\phi]}{[\phi]}$

◆ Preliminaries

Parameters

$:=$ 从右向左赋值

$=:$ 从左向右赋值

$\leftarrow$ 从分布中从右向左采样

$b_{*,*}$ 矩阵

| x | x的字长

| **y** | 向量y中元素的个数

λ_c和λ_s 分别表示计算和统计安全参数

κ为表示椭圆曲线阶数域元素所需的位数

Preliminaries

Modules

$F_{RelaxedKeyGen}$ F_{Zero} → n party

+

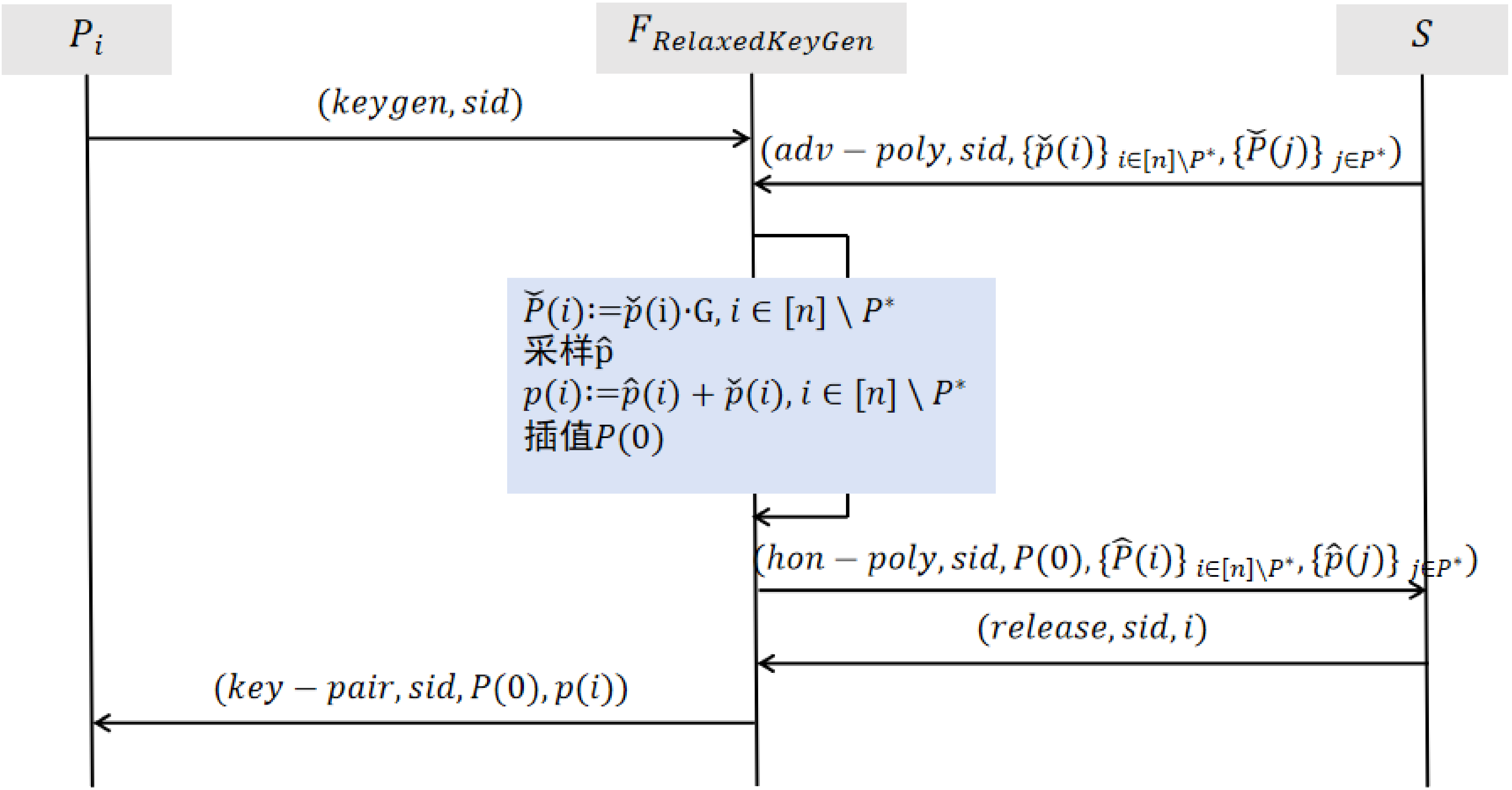
F_{Com} F_{RVOLE} → 2 party

↓

F_{ECDSA} n party

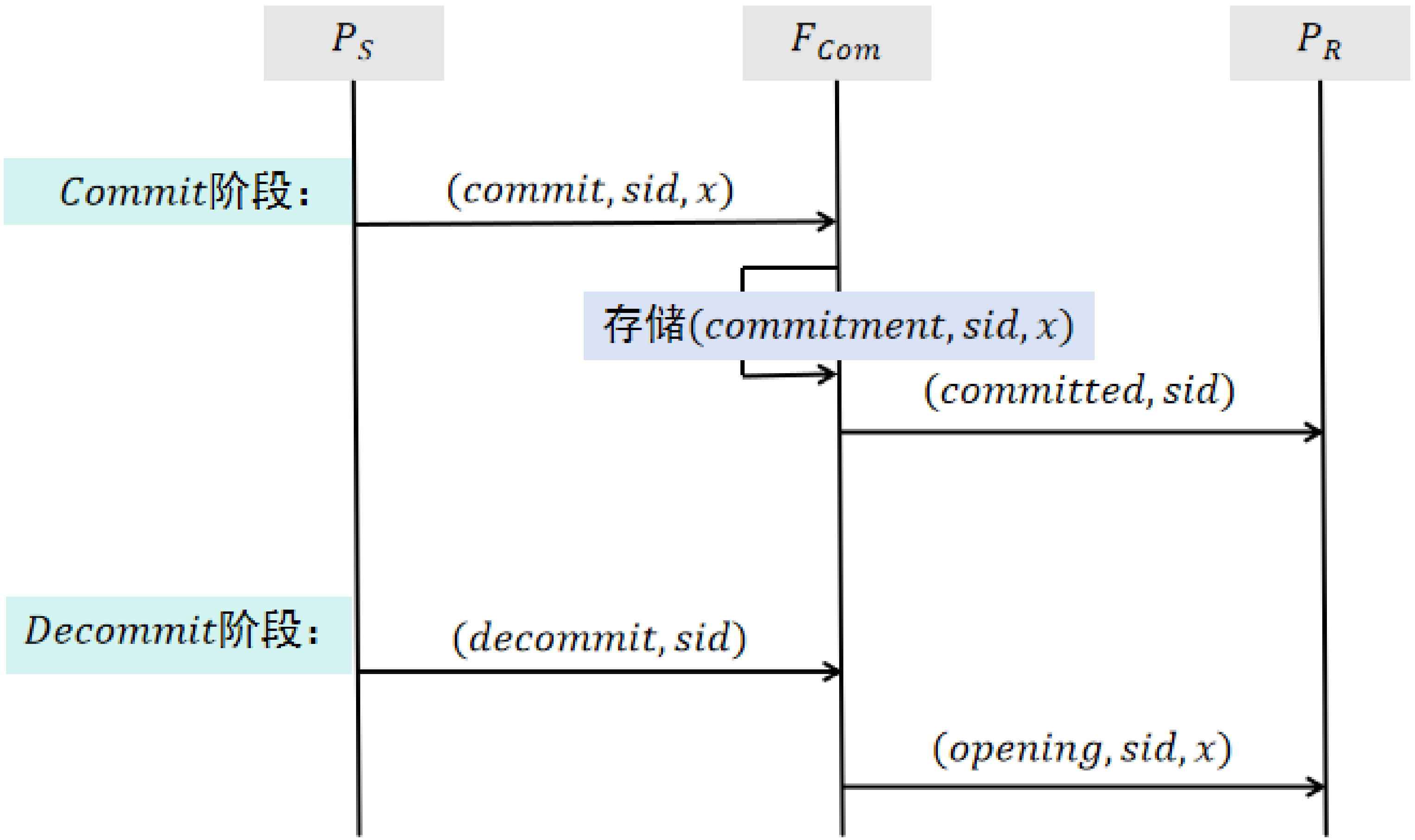
◆ Preliminaries

$F_{RelaxedKeyGen}(G, n, t)$: Relaxd Dlog Keygen



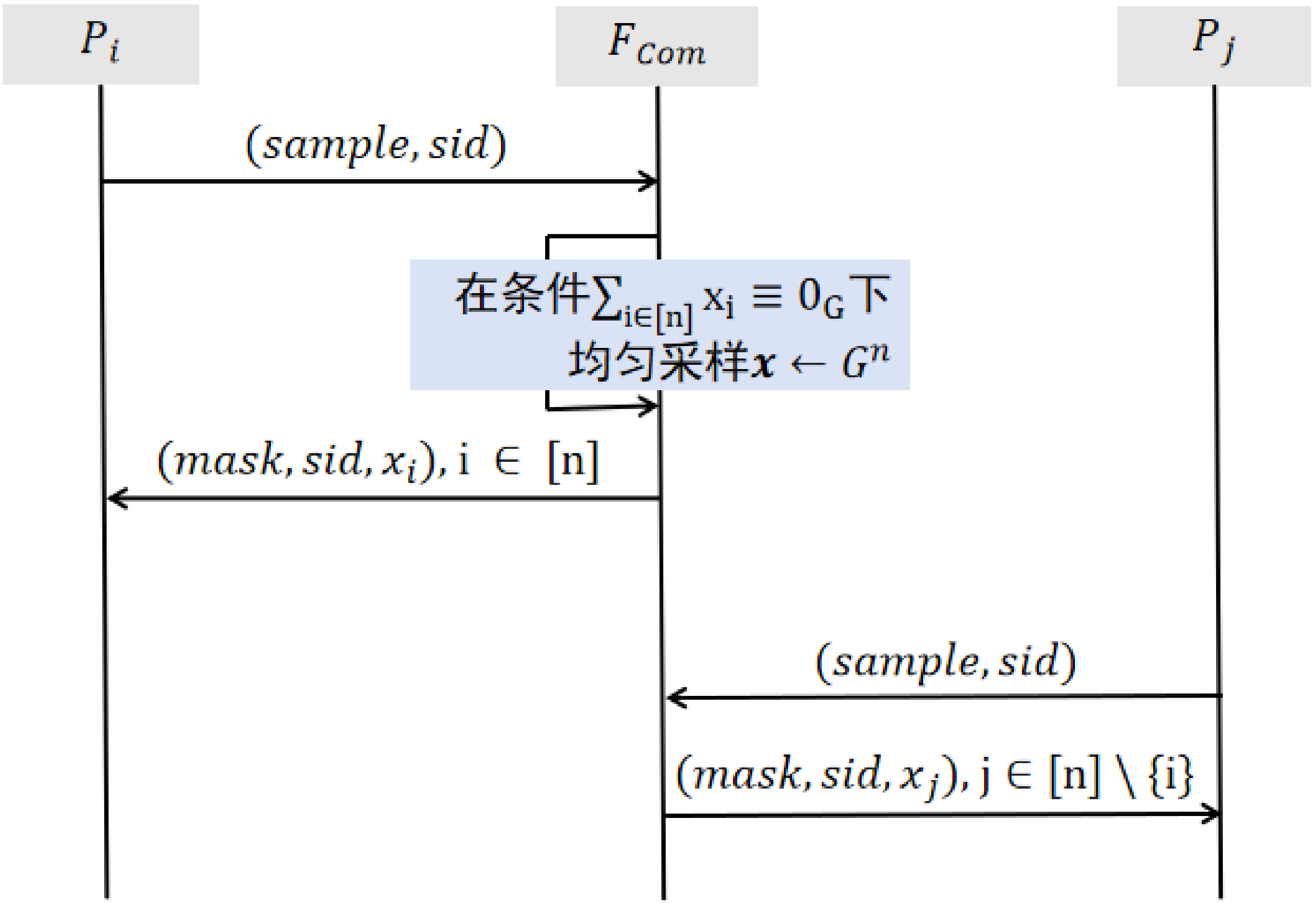
◆ Preliminaries

F_{Com}: Commitment



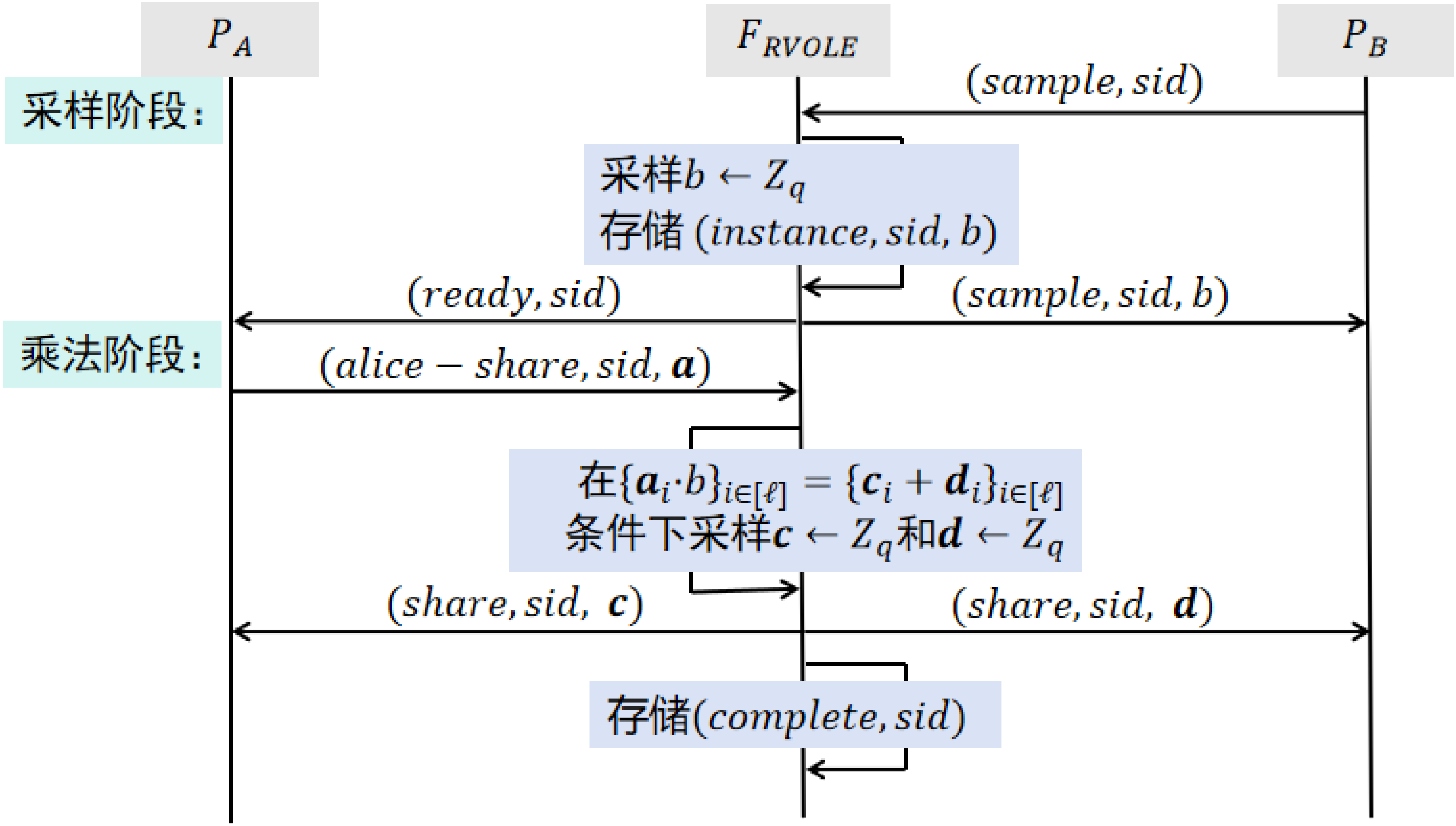
◆ Preliminaries

$F_{Zero}(G,n)$: Zero–Sharing Sampling



◆ Preliminaries

$F_{RVOLE}(q,\ell)$: Random Vector OLE



t-Party Three-Round Threshold ECDSA

Framework

Input $[sk]$

Local — Sample $[r][\phi]$

Round 1 / Round 2 — Establish $R = [r] \cdot G$

Securely Compute $[r \cdot \phi]$ $[sk \cdot \phi]$ ← Many Methods

Verify Consistency ← New Work

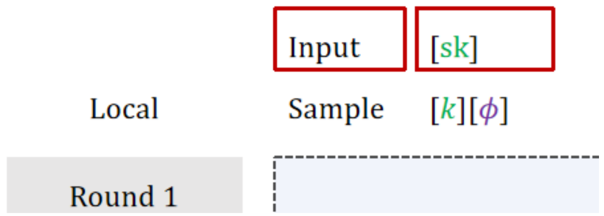
Local — $w = e \cdot [\phi] + r_x \cdot [sk \cdot \phi]$

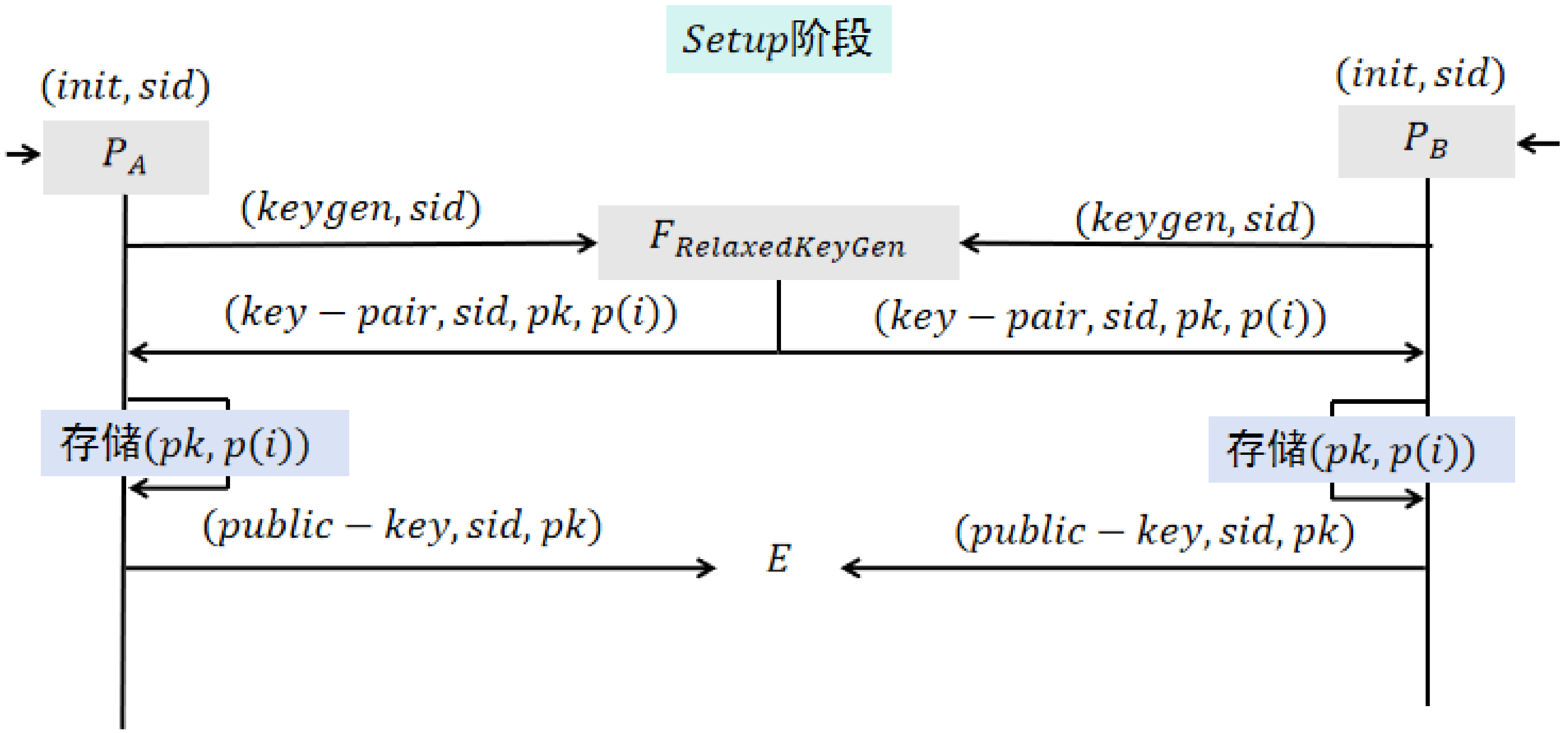
Round 3 — Reveal $w, u = r \cdot [\phi]$

Output $(w/u, R)$

◆ t-Party Three-Round Threshold ECDSA

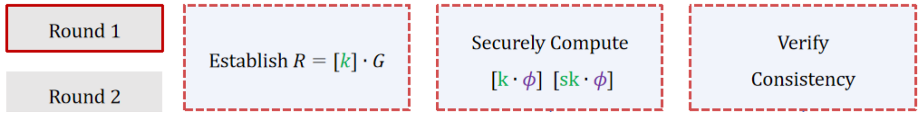
The Basic Three-Round Protocol

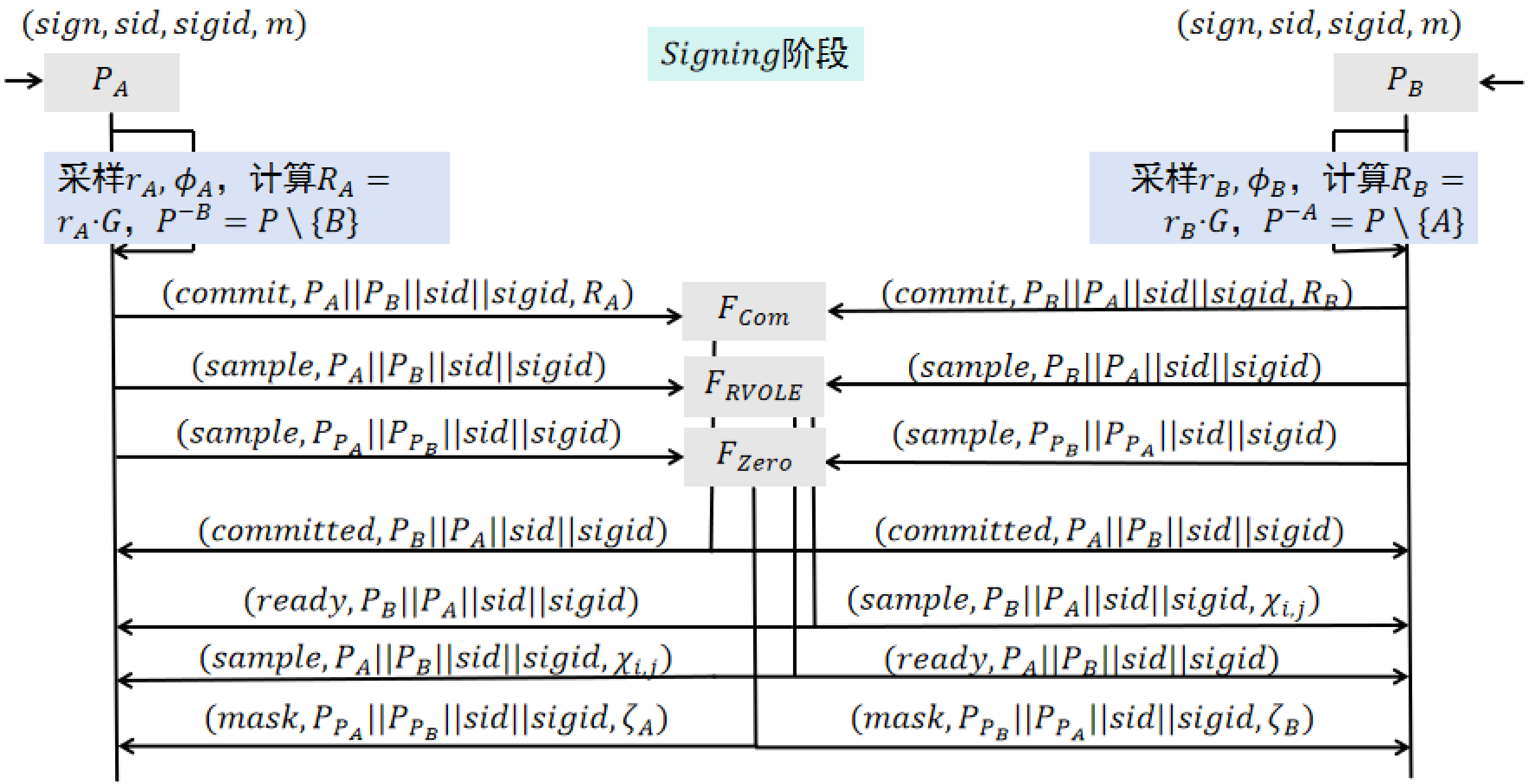




◆ t-Party Three-Round Threshold ECDSA

The Basic Three-Round Protocol





◆ t-Party Three-Round Threshold ECDSA

The Basic Three-Round Protocol

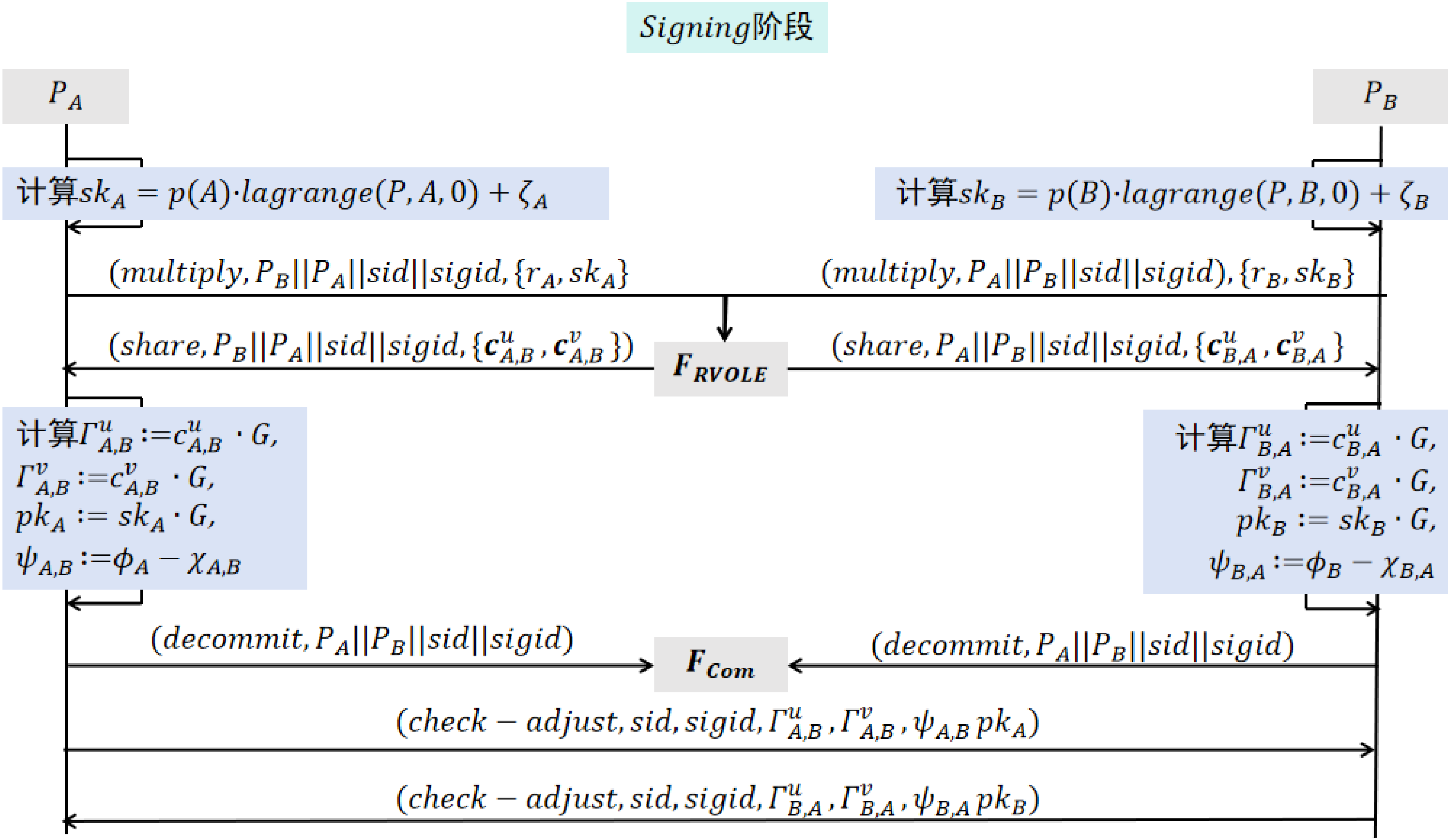
Round 1

Round 2

Establish $R = [k] \cdot G$

Securely Compute $[k \cdot \phi]$ $[sk \cdot \phi]$

Verify Consistency



◆ t-Party Three-Round Threshold ECDSA

The Basic Three-Round Protocol

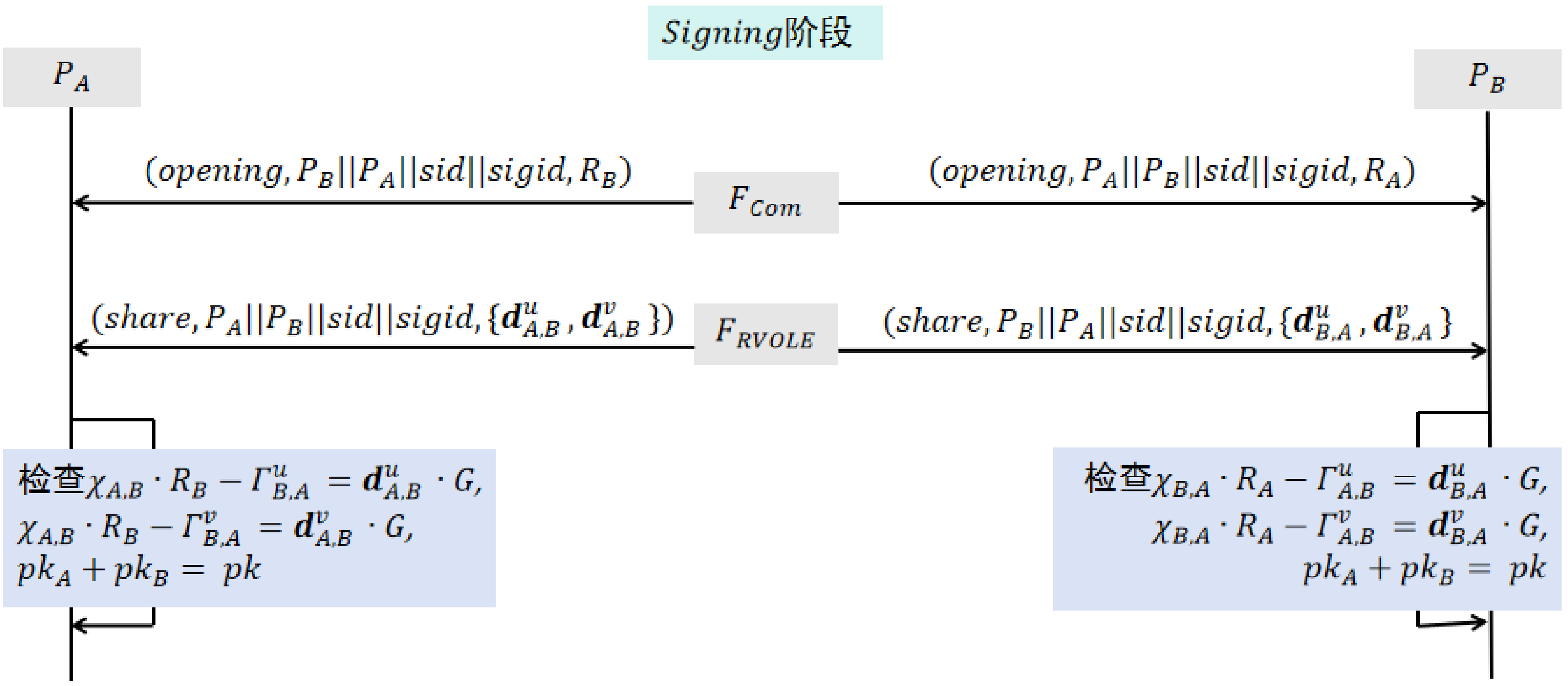
Round 1

Round 2

Establish $R = [k] \cdot G$

Securely Compute $[k \cdot \phi]$ $[sk \cdot \phi]$

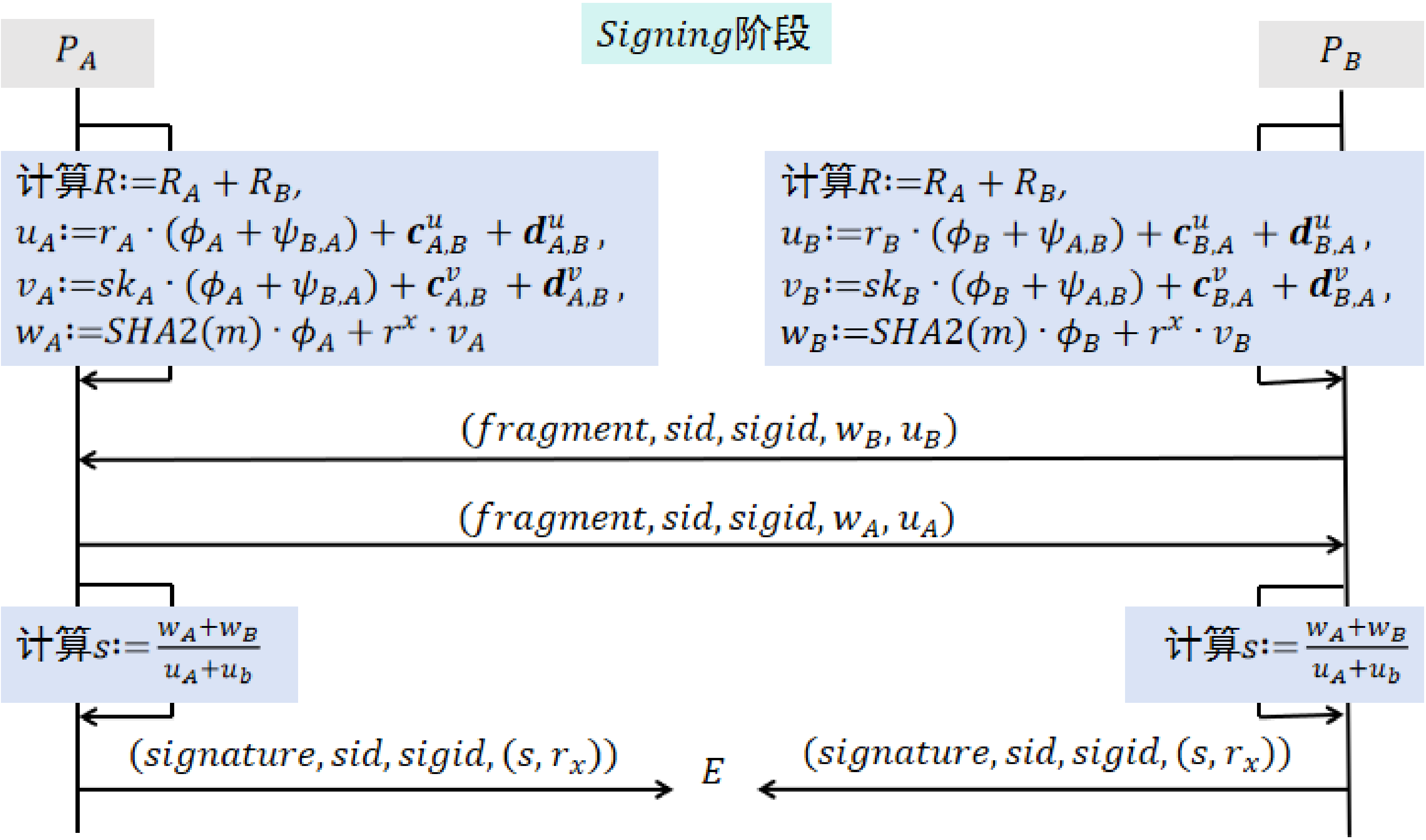
Verify Consistency



t-Party Three-Round Threshold ECDSA

The Basic Three-Round Protocol

Local	$\alpha = e \cdot [\phi] + r_x \cdot [sk \cdot \phi]$
Round 3	Reveal $\alpha, \beta = k \cdot [\phi]$
	Output $(\alpha/\beta, R)$



t-Party Three-Round Threshold ECDSA

Pipelining and Presigning

采样r_A, ϕ_A
计算R_A, P^{-B}

ROUND1:
send $(commit, R_A)$, sample

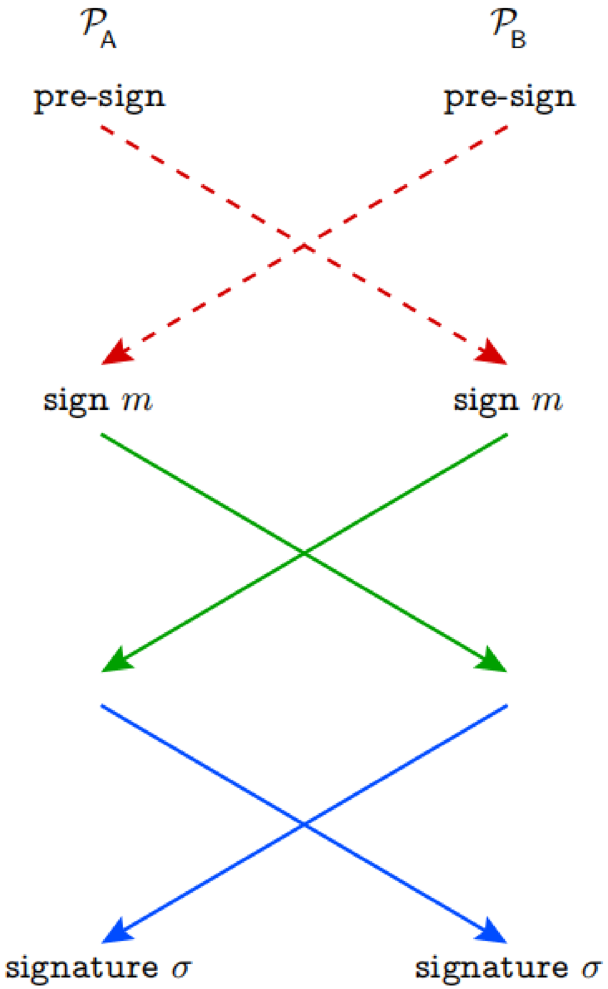
收到committed, ready, $(sample, \chi_{i,j})$, $(mask, \zeta_A)$
计算$sk_A, \Gamma^u_{A,B}, \Gamma^v_{A,B}, pk_A, \psi_{A,B}$

ROUND2:
send $(multiply, \{r_A, sk_A\})$, decommit, $(check-adjust, \Gamma^u_{A,B}, \Gamma^v_{A,B}, \psi_{A,B}\ pk_A)$

收到$(opening, R_B)$, $(share, \{d^u_{A,B}, d^v_{A,B}\})$
检查，计算R, u_A, v_A, w_A

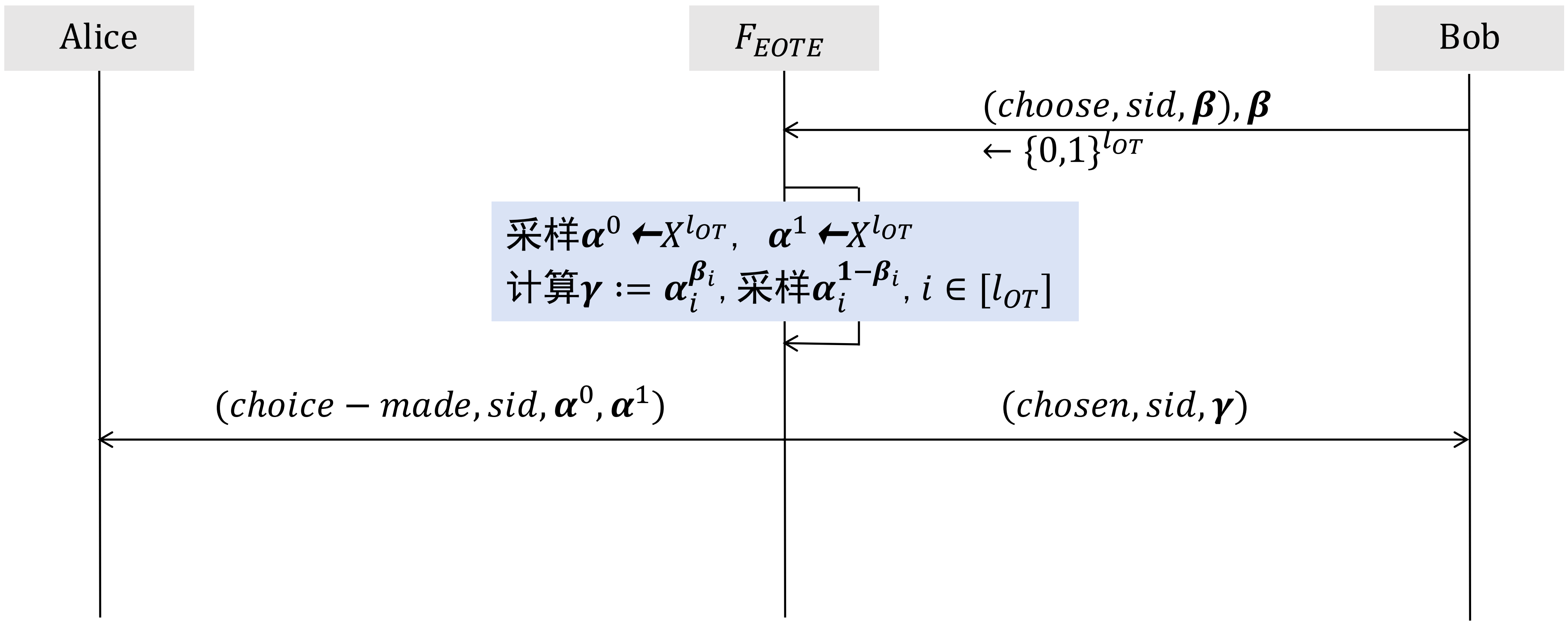
ROUND3:
send $(fragment, w_A, u_A)$

计算s，生成签名



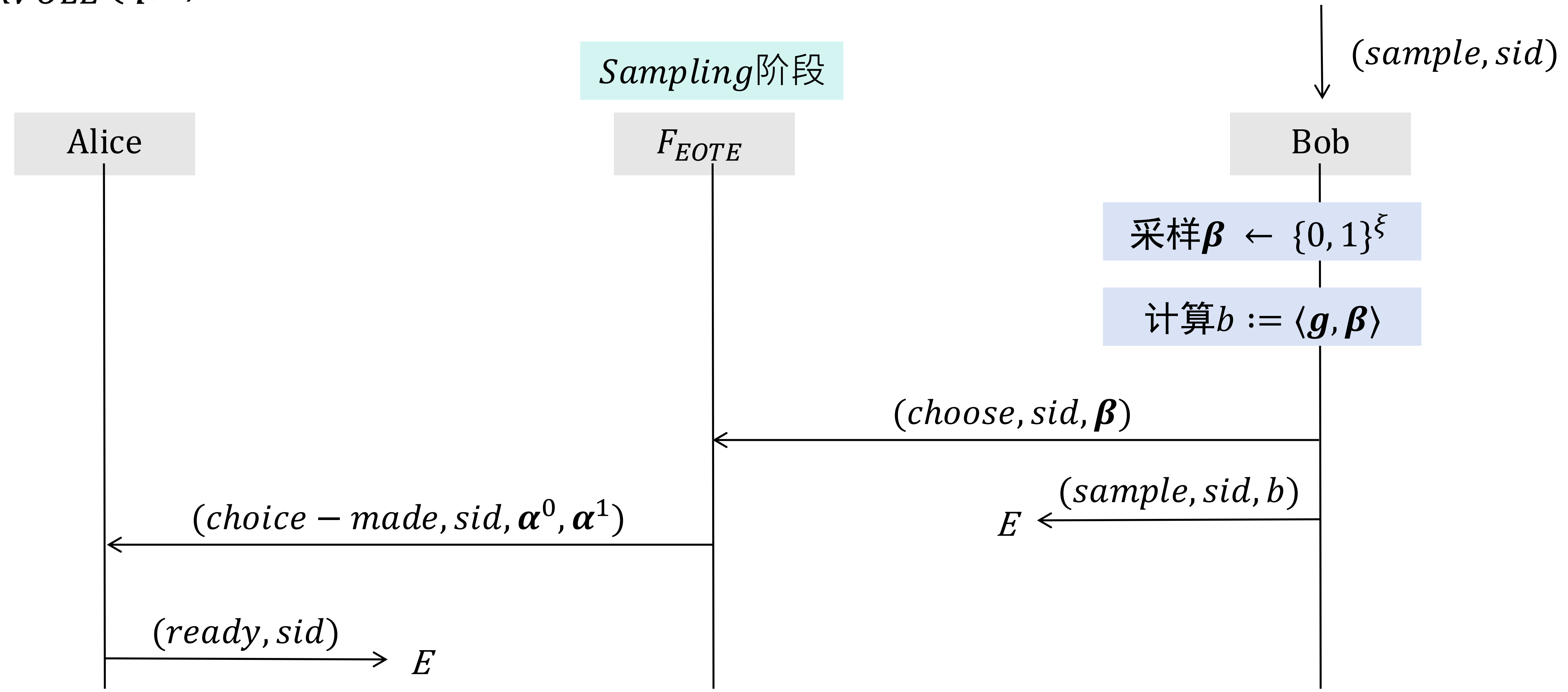
◆ Random Vector OLE from Random OT

$F_{EOTE}(X, l_{OT})$: Endemic OT Extension



Random Vector OLE from Random OT

$F_{RVOLE}(q, l)$: OT-Based Random Vector OLE



◆ Random Vector OLE from Random OT

$F_{RVOLE}(q,l)$: OT-Based Random Vector OLE

Alice ← $(multiply, sid, \boldsymbol{a})$ | $Multiplication$阶段 | Bob

$$\mathbf{c} := \left\{ -\sum_{j\in[\xi]} \mathbf{g}_j \cdot \boldsymbol{\alpha}^0_{j,i} \right\}_{i\in[\ell]}$$

$$\hat{\mathbf{a}} \leftarrow \mathbb{Z}_q^{\rho}$$

$$\tilde{\mathbf{a}} := \left\{ \{\boldsymbol{\alpha}^0_{j,i} - \boldsymbol{\alpha}^1_{j,i} + \mathbf{a}_i\}_{i\in[\ell]} \,\|\, \{\boldsymbol{\alpha}^0_{j,\ell+k} - \boldsymbol{\alpha}^1_{j,\ell+k} + \hat{\mathbf{a}}_i\}_{k\in[\rho]} \right\}_{j\in[\xi]}$$

$$\boldsymbol{\theta} := \mathsf{RO}_{\mathbb{Z}_q^{\ell\times\rho}}(\mathsf{sid}, \tilde{\mathbf{a}})$$

$$\boldsymbol{\eta} := \left\{ \hat{\mathbf{a}}_k + \sum_{i\in[\ell]} \boldsymbol{\theta}_{i,k} \cdot \mathbf{a}_i \right\}_{k\in[\rho]}$$

$$\boldsymbol{\mu} := \left\{ \left\{ \boldsymbol{\alpha}^0_{j,\ell+k} + \sum_{i\in[\ell]} \boldsymbol{\theta}_{k,k} \cdot \boldsymbol{\alpha}^0_{j,i} \right\}_{k\in[\rho]} \right\}_{j\in[\xi]}$$

$$\mu := \mathsf{RO}_{\{0,1\}^{2\lambda_c}}(\mathsf{sid}, \boldsymbol{\mu})$$

Alice → Bob: $(multiply, sid, \tilde{\boldsymbol{a}}, \boldsymbol{\eta}, \mu)$

Alice → E: $(share, sid, \boldsymbol{c})$

◆ Random Vector OLE from Random OT

$F_{RVOLE}(q,l)$: OT-Based Random Vector OLE

*Multiplication*阶段

Bob

$$\boldsymbol{\theta} := \mathsf{RO}_{\mathbb{Z}_q^{\ell\times\rho}}(\mathsf{sid}, \tilde{\mathbf{a}})$$

$$\dot{\mathbf{d}} := \left\{\left\{\boldsymbol{\gamma}_{j,i} + \boldsymbol{\beta}_j \cdot \tilde{\mathbf{a}}_{j,i}\right\}_{i\in[\ell]}\right\}_{j\in[\xi]}$$

$$\hat{\mathbf{d}} := \left\{\left\{\boldsymbol{\gamma}_{j,\ell+k} + \boldsymbol{\beta}_j \cdot \tilde{\mathbf{a}}_{j,\ell+k}\right\}_{k\in[\rho]}\right\}_{j\in[\xi]}$$

$$\boldsymbol{\mu}' := \left\{\left\{\hat{\mathbf{d}}_{j,k} + \sum_{i\in[\ell]} \boldsymbol{\theta}_{i,k} \cdot \dot{\mathbf{d}}_{j,i} - \boldsymbol{\beta}_j \cdot \boldsymbol{\eta}_k\right\}_{k\in[\rho]}\right\}_{j\in[\xi]}$$

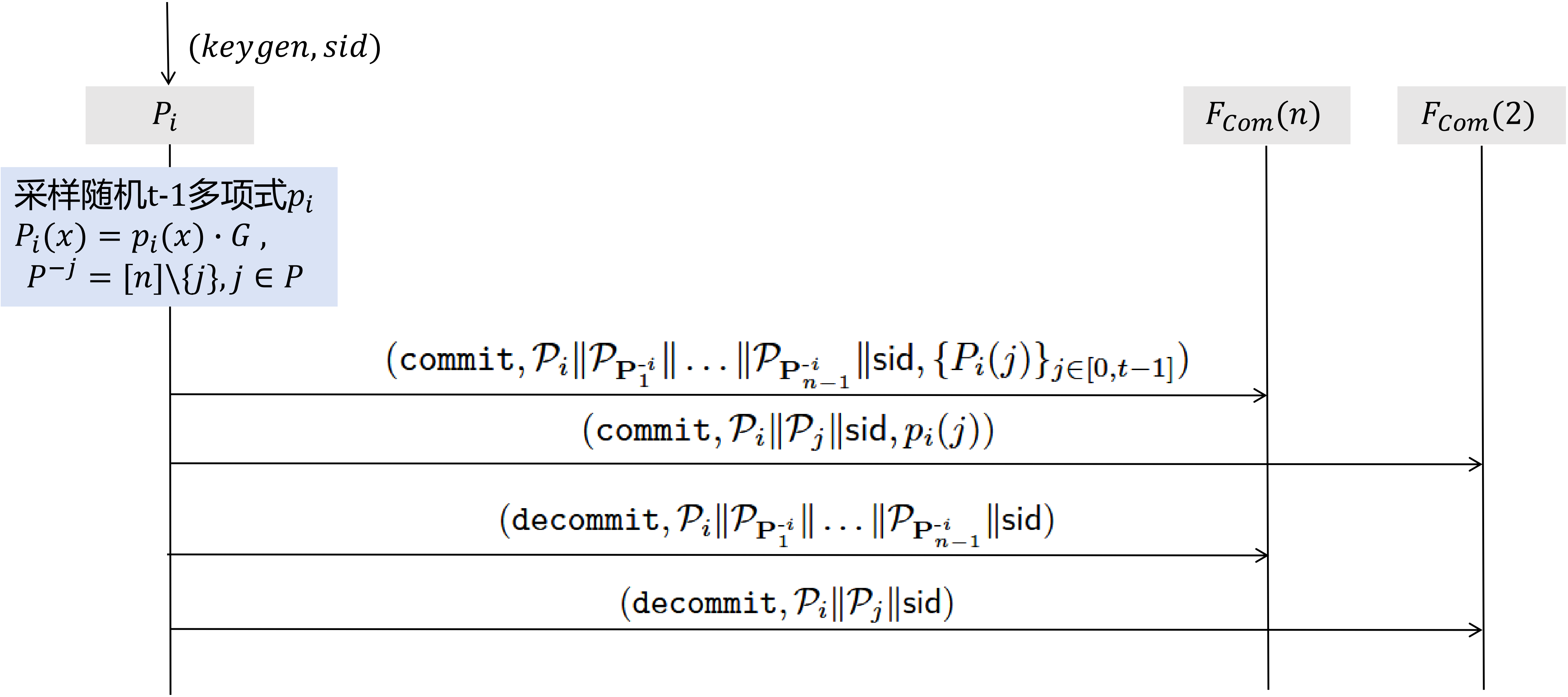
检查 $\mu = \mathsf{RO}_{\{0,1\}^{2\lambda_c}}(\mathsf{sid}, \boldsymbol{\mu}')$

$$\mathbf{d} := \left\{\sum_{j\in[\xi]} \mathbf{g}_j \cdot \dot{\mathbf{d}}_{j,i}\right\}_{i\in[\ell]}$$

E ← $(share, sid, \boldsymbol{d})$

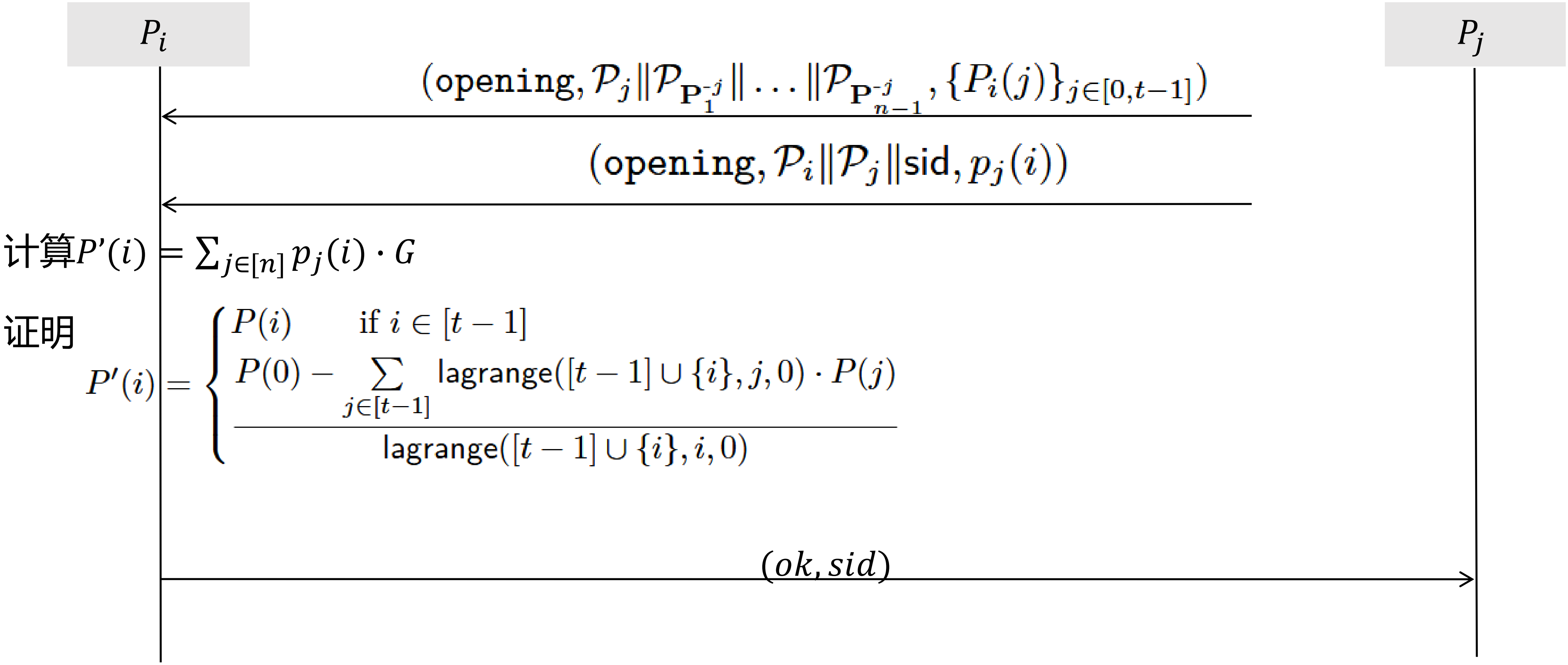
◆ Relaxed Threshold Key Generation

$\pi_{RelaxedKeyGen}(G, n, t)$: Relaxed DLog Keygen



◆ Relaxed Threshold Key Generation

$\pi_{RelaxedKeyGen}(G,n,t)$: Relaxed DLog Keygen



◆ Analytical Efficiency

Communication Cost

Commit(2)

commit → 2 λc
decommit(x) → 2 λc + x

Commit(n)

commit → (n-1) [2 λc + 2 λc + x] + 2n λc

Zero

t-1 commit+decommit(λc)

OT

$$\mathsf{EOTECost}(\lambda_c, \ell_{\mathsf{OTE}}) \mapsto \left(\frac{3}{2} + \frac{1}{2k_{\mathsf{SSOT}}}\right) \cdot ({\lambda_c}^2 + \lambda_c) + \frac{\lambda_c \cdot \ell_{\mathsf{OTE}}}{2k_{\mathsf{SSOT}}}$$

VOLE

$$\mathsf{VOLECost}(\lambda_c, \lambda_s, \kappa, \ell) \mapsto$$
$$\mathsf{EOTECost}(\lambda_c, \kappa + 2\lambda_s) + (\kappa/2 + \lambda_s) \cdot (\ell + 1) \cdot \kappa + \kappa/2 + \lambda_c$$

$$\mathsf{VOLESetupCost}(\lambda_c, \lambda_s, \kappa, |G|) \mapsto \mathsf{EOTCost}(|G|, \lambda_c) + \lambda_c/2$$

◆ Analytical Efficiency

Communication Cost

采样r_A, ϕ_A
计算R_A, P^{-B}

ROUND1:
send ($commit, R_A$), sample

收到committed, ready, ($sample, \chi_{i,j}$), ($mask, \zeta_A$)
计算$sk_A, \Gamma^u_{A,B}, \Gamma^v_{A,B}, pk_A, \psi_{A,B}$

ROUND2:
send ($multiply, \{r_A, sk_A\}$), decommit, ($check - adjust, \Gamma^u_{A,B}, \Gamma^v_{A,B}, \psi_{A,B}\, pk_A$)

收到($opening, R_B$), ($share, \{d^u_{A,B}, d^v_{A,B}\}$)
检查，计算R, u_A, v_A, w_A

ROUND3:
send ($fragment, w_A, u_A$)

计算s，生成签名

λ_c和λ_s 分别表示计算和统计安全参数

κ为表示椭圆曲线阶数域元素所需的位数

RelaxedKeyGen

$$\mathsf{KeyGenCost}(n, \lambda_c, \kappa, |G|) \mapsto (n-1)\cdot(10\lambda_c + t\cdot|G| + \kappa)$$

Sign

$$\mathsf{SignCost}(t, \lambda_c, \lambda_s, \kappa, |G|) \mapsto (t-1)\cdot(4\lambda_c + 3\kappa + 4|G| + 2\cdot\mathsf{VOLECost}(\lambda_c, \lambda_s, \kappa, 2))$$

◆ Analytical Efficiency

Computation Cost

RelaxedKeyGen	2t EC
VOLE	$6\lambda c(n-1)$ EC
Sign	6t-2 EC

◆ Analytical Efficiency

Compared with DKLs

在所有情况下，假设$\kappa = 2\lambda c$，$\lambda s = 80$

$\lambda c = 256$，$\lambda s = 80$

	2-of-n	t-of-n	EC
DKLS	116.4 KiB	$(t-1) \cdot 88.3$ KiB	6
Our	49.7 KiB	$(t-1) \cdot 49.7$ KiB	6t-2

Analytical Efficiency

Bandwidth Costs

在所有情况下，假设$\kappa = 2\lambda c$，$\lambda s = 80$

λ_c	128	192	256
κ	256	384	512
$\|G\|$	264	392	520
Setup	$(n-1)\cdot 137232$	$(n-1)\cdot 304144$	$(n-1)\cdot 536592$
Signing (our VOLE)	$(t-1)\cdot 406752$	$(t-1)\cdot 812864$	$(t-1)\cdot 1354144$
Signing (HMRT22)	$(t-1)\cdot 392544$	$(t-1)\cdot 742400$	$(t-1)\cdot 1194656$

Thanks