Threshold ECDSA from ECDSA Assumptions: The Multiparty Case - 2019

Doerner, Kondi, Lee, shelat (DKLs)

Selman Pınar

TÜBİTAK - BİLGEM

November 21, 2025

Overview

ECDSA

Algorithm 1. $\mathsf{Gen}(1^\kappa)$**:**

1) Uniformly choose a secret key $\mathsf{sk} \leftarrow \mathbb{Z}_q$.
2) Calculate the public key as $\mathsf{pk} := \mathsf{sk} \cdot G$.
3) Output $(\mathsf{pk}, \mathsf{sk})$.

Algorithm 2. $\mathsf{Sign}(\mathsf{sk} \in \mathbb{Z}_q, m \in \{0,1\}^*)$**:**

1) Uniformly choose an instance key $k \leftarrow \mathbb{Z}_q$.
2) Calculate $(r_x, r_y) = R := k \cdot G$.
3) Calculate
$$\mathsf{sig} := \frac{H(m) + \mathsf{sk} \cdot r_x}{k}$$
4) Output $\sigma := (\mathsf{sig} \mod q, r_x \mod q)$.

Algorithm 3. $\mathsf{Verify}(\mathsf{pk} \in \mathbb{G}, m, \sigma \in (\mathbb{Z}_q, \mathbb{Z}_q))$**:**

1) Parse σ as (sig, r_x).
2) Calculate
$$(r'_x, r'_y) = R' := \frac{H(m) \cdot G + r_x \cdot \mathsf{pk}}{\mathsf{sig}}$$
3) Output 1 if and only if $(r'_x \mod q) = (r_x \mod q)$.

(t, n)-ECDSA Idea

ECDSA sign is

$$\mathsf{sig} = \frac{1}{k} \cdot H(m) + \frac{\mathsf{sk}}{k} \cdot r_x$$

(t, n)-ECDSA Idea

ECDSA sign is

$$\text{sig} = \frac{1}{k} \cdot H(m) + \frac{\text{sk}}{k} \cdot r_x$$

We will split the secret parameters of signature into t additive pieces such that

$$\frac{1}{k} = \sum_{i \in P} v_i \qquad \frac{sk}{k} = \sum_{i \in P} w_i \quad \text{where } |P| = t$$

(t, n)-ECDSA Idea

ECDSA sign is

$$\text{sig} = \frac{1}{k} \cdot H(m) + \frac{\text{sk}}{k} \cdot r_x$$

We will split the secret parameters of signature into t additive pieces such that

$$\frac{1}{k} = \sum_{i \in P} v_i \qquad \frac{sk}{k} = \sum_{i \in P} w_i \quad \text{where } |P| = t$$

The i'th party knows v_i, w_i and no other party knows these values.

(t, n)-ECDSA Idea

ECDSA sign is

$$\text{sig} = \frac{1}{k} \cdot H(m) + \frac{\text{sk}}{k} \cdot r_x$$

We will split the secret parameters of signature into t additive pieces such that

$$\frac{1}{k} = \sum_{i \in P} v_i \qquad \frac{sk}{k} = \sum_{i \in P} w_i \quad \text{where } |P| = t$$

The i'th party knows v_i, w_i and no other party knows these values.
i'th party calculates

$$\text{sig}_i = v_i \cdot H(m) + w_i \cdot r_x$$

(t, n)-ECDSA Idea

ECDSA sign is

$$\text{sig} = \frac{1}{k} \cdot H(m) + \frac{\text{sk}}{k} \cdot r_x$$

We will split the secret parameters of signature into t additive pieces such that

$$\frac{1}{k} = \sum_{i \in P} v_i \qquad \frac{sk}{k} = \sum_{i \in P} w_i \quad \text{where } |P| = t$$

The i'th party knows v_i, w_i and no other party knows these values.
i'th party calculates

$$\text{sig}_i = v_i \cdot H(m) + w_i \cdot r_x$$

and the original signature will be

$$\text{sig} = \sum_{i \in P} \text{sig}_i$$

(t, n)-ECDSA Idea

ECDSA sign is

$$\text{sig} = \frac{1}{k} \cdot H(m) + \frac{\text{sk}}{k} \cdot r_x$$

We will split the secret parameters of signature into t additive pieces such that

$$\frac{1}{k} = \sum_{i \in P} v_i \qquad \frac{sk}{k} = \sum_{i \in P} w_i \quad \text{where } |P| = t$$

The i'th party knows v_i, w_i and no other party knows these values.
i'th party calculates

$$\text{sig}_i = v_i \cdot H(m) + w_i \cdot r_x$$

and the original signature will be

$$\text{sig} = \sum_{i \in P} \text{sig}_i$$

Remark

Nobady can learn the secrets sk and k in this process, only the signature is created.

OT functionality

Sender		**Receiver**
Input: (m_0, m_1)		Input: $b \in \{0, 1\}$
	$\xrightarrow{(m_0, m_1)}$	
Output: nothing, $\perp$		Output: m_b
		Does not know: m_{1-b}

The Simplest Protocol for OT - *Tung Chou and Claudio Orlandi*

Sender		**Receiver**
Input: (M_0, M_1)		Input: c
Output: none		Output: M_c
$a \leftarrow \mathbb{Z}_p$		$b \leftarrow \mathbb{Z}_p$
	$\xrightarrow{A = g^a}$	
		if $c = 0$: $B = g^b$
		if $c = 1$: $B = Ag^b$
	$\xleftarrow{B}$	
$k_0 = H\left(B^a\right)$		$k_R = H(A^b)$
$k_1 = H\left(\left(\frac{B}{A}\right)^a\right)$		
	$e_0 \leftarrow E_{k_0}(M_0)$	
	$e_1 \leftarrow E_{k_1}(M_1)$	
	$\longrightarrow$	
		$M_c = D_{k_R}(e_c)$

Beaver MtA Tecnique with OT

Protokol:

Bob	Alice
b	$a = (a_{\rho-1} \dots a_0)_2$

$$s_0, s_1, \dots, s_{\rho-1} \xleftarrow{\$} R$$

$$\begin{array}{lllll}
t_0^0 = s_0, & t_0^1 = b + s_0 & \xrightarrow{t_0^0,\ t_0^1} & \text{choose} & t_0^{a_0} \\
t_1^0 = s_1, & t_1^1 = 2^1 b + s_1 & \xrightarrow{t_1^0,\ t_1^1} & \text{choose} & t_1^{a_1} \\
t_2^0 = s_2, & t_2^1 = 2^2 b + s_2 & \xrightarrow{t_2^0,\ t_2^1} & \text{choose} & t_2^{a_2} \\
\vdots & \vdots & & & \vdots \\
t_i^0 = s_i, & t_i^1 = 2^i b + s_i & \xrightarrow{t_i^0,\ t_i^1} & \text{choose} & t_i^{a_i}
\end{array}$$

$$y = -\sum_{i=0}^{\rho-1} s_i \qquad\qquad x = \sum_{i=0}^{\rho-1} t_i^{a_i}$$

İspat:

$$\begin{aligned}
x + y &= \sum_{i=0}^{\rho-1} t_i^{a_i} - \sum_{i=0}^{\rho-1} s_i \\
&\equiv \sum_{i=0}^{\rho-1} (a_i \cdot 2^i b + s_i) - \sum_{i=0}^{\rho-1} s_i \\
&\equiv b \sum_{i=0}^{\rho-1} a_i \cdot 2^i \\
&\equiv ab
\end{aligned}$$

VSOT

Sender		Receiver
Two messages α^0, α^1		Choice bit $\omega \in \{0,1\}$
$b \leftarrow \mathbb{Z}_q$ $B := b \cdot G$	$\xrightarrow{B}$	DL-ZKProve(B)
$\rho^0 := H(b \cdot A)$ $\rho^1 := H(b(A-B))$	$\xleftarrow{A}$	$a \leftarrow \mathbb{Z}_q$ $A := a \cdot G + \omega \cdot B$ $\rho^\omega := H(a \cdot B)$
Compute a challenge: $\varepsilon := H(H(\rho^0)) \oplus H(H(\rho^1))$	$\xrightarrow{\varepsilon}$	
	$\xleftarrow{\rho'}$	Compute a response $\rho' := H(H(\rho^\omega)) \oplus (\omega \cdot \varepsilon)$
Abort if $\rho' \neq H(H(\rho^0))$		
	$\xrightarrow{H(\rho^0), H(\rho^1)}$	Abort if $H(\rho^\omega)$, Does not match the one calculated itself
		Abort if $\varepsilon \neq H(H(\rho^0)) \oplus H(H(\rho^1))$
$\tilde{\alpha}^0 := \alpha^0 + \rho^0$ $\tilde{\alpha}^1 := \alpha^1 + \rho^1$	$\xrightarrow{\tilde{\alpha}^0, \tilde{\alpha}^1}$	$\alpha^\omega = \tilde{\alpha}^\omega - \rho^\omega$

Public Key, Pad Transfer, Verification, Message Transfer

Sender

correlation:

$\alpha \in G_1 \times G_2 \times \cdots \times G_\ell$

$t_S \xleftarrow{\$} G_1 \times G_2 \times \cdots \times G_\ell \longleftarrow$

Output t_S

$$\mathcal{F}^{\ell}_{COT}$$

Receiver

$w \in \{0,1\}^\ell$: choice

if $w_i = 0 \longrightarrow -t_{S_i}$

if $w_i = 1 \longrightarrow \alpha_i - t_{S_i}$

which means:

$t_R : \{w_i \cdot \alpha_i - t_{S_i}\}_{i \in [1,\ell]}$

Output t_R

$$t_S + t_R = w * \alpha$$

KOS (Keller, Orsini, Scholl) Setup

- KOS Setup ve KOS Extension protokolleri yukarıdaki $\mathcal{F}^{\ell}_{\mathsf{COTe}}$ fonksiyonunu gerçekleştirir.

Alice		Bob
$\boldsymbol{\Delta} \leftarrow \{0,1\}^{\kappa}$		$s_i^0 \leftarrow \mathbb{Z}_q$
	$\xleftarrow{\text{VSOT } s_i^0,\, s_i^1}$	$s_i^1 \leftarrow \mathbb{Z}_q$
For all i take: $s_i^{\Delta_i}$		
output:		output:
$\boldsymbol{s}^{\boldsymbol{\Delta}} = \{s_i^{\Delta_i}\}_{i \in [1,\kappa]}$		$\boldsymbol{s}^{\mathbf{0}} = \{s_i^0\}_{i \in [1,\kappa]}$
		$\boldsymbol{s}^{\mathbf{1}} = \{s_i^1\}_{i \in [1,\kappa]}$

KOS Extension

Alice		Bob
inputs: $\boldsymbol{\alpha} \in G_1 \times G_2 \times \cdots \times G_\ell$ $\boldsymbol{\Delta}$ and $s^{\boldsymbol{\Delta}}$		inputs: $\boldsymbol{\omega} \in \{0,1\}^\ell$, $\boldsymbol{s^0}$ and $\boldsymbol{s^1}$
		$\gamma^{\text{ext}} \leftarrow \{0,1\}^{\kappa^{\text{OT}}}$ $\mathbf{w} := \boldsymbol{\omega} \| \gamma^{\text{ext}}$ $w := \text{Bits}^{-1}(\mathbf{w}) \in \mathbb{Z}_{2^\kappa}^{\ell'}$
$\mathbf{v}^{\boldsymbol{\Delta}} := (\text{PRG}(s_i^{\Delta_i}))_{i\in[1,\kappa]}$	(PRG: $\mathbb{Z}_q^\kappa \to \mathbb{Z}_{2^{\ell'}}$)	$\mathbf{v}^0 := (\text{PRG}(s_i^0))_{i\in[1,\kappa]}$ $\mathbf{v}^1 := (\text{PRG}(s_i^1))_{i\in[1,\kappa]}$
		$\psi := \text{transpose}(\mathbf{v}^0) \in \mathbb{Z}_{2^\kappa}^{\ell'}$ $\mathbf{u} := \{v_i^0 \oplus v_i^1 \oplus w\}_{i\in[1,\kappa]}$ $\boldsymbol{\chi} := \{H(j\|\mathbf{u})\}_{j\in[1,\ell']}$ $w' := \bigoplus_{j\in[1,\ell']} w_j \cdot \chi_j$ $\psi' := \bigoplus_{j\in[1,\ell']} \psi_j \cdot \chi_j$
	$\xleftarrow{(w', \psi', \mathbf{u})}$	
$\mathbf{z} := \{v_i^{\Delta_i} \oplus (\Delta_i \cdot u_i)\}_{i\in[1,\kappa]}$ $\zeta := \text{transpose}(\mathbf{z}) \in \mathbb{Z}_{2^\kappa}^{\ell'}$ $\boldsymbol{\chi} := \{H(j\|\mathbf{u})\}_{j\in[1,\ell']}$ $z' := \bigoplus_{j\in[1,\ell']} \zeta_j \cdot \chi_j$ If $z' \neq \psi' \oplus (\Delta \cdot w')$ abort		
$\mathbf{t}_A := \{H(j\|\zeta_j)\}_{j\in[1,\ell]}$ $\boldsymbol{\tau} := \{H(j\|(\zeta_j \oplus \Delta)) - \mathbf{t}_{Aj} + \boldsymbol{\alpha}_j\}_{j\in[1,\ell]}$		
	$\xrightarrow{\boldsymbol{\tau}}$	
		$\mathbf{t}_B := \left\{ \begin{cases} -H(j\|\psi_j) & \text{if } \mathbf{w}_j = 0 \\ \tau_j - H(j\|\psi_j) & \text{if } \mathbf{w}_j = 1 \end{cases} \right\}_{j\in[1,\ell]}$
output: $\mathbf{t}_A$		output $\mathbf{t}_B$

$$\mathbf{t}_A + \mathbf{t}_B = \boldsymbol{\omega} * \boldsymbol{\alpha}$$

Why Consistency Check works?

We need to show that: $z' = \psi' \oplus (\Delta \cdot w')$

Why Consistency Check works?

We need to show that: $z' = \psi' \oplus (\Delta \cdot w')$
Proof:

Why Consistency Check works?

We need to show that: $z' = \psi' \oplus (\Delta \cdot w')$
Proof:
The defination of the z, we have:

$$z_i = v_i^{\Delta_i} \oplus (\Delta_i \cdot u_i)$$

Why Consistency Check works?

We need to show that: $z' = \psi' \oplus (\Delta \cdot w')$
Proof:
The defination of the z, we have:

$$z_i = v_i^{\Delta_i} \oplus (\Delta_i \cdot u_i)$$

and the defination of the u_i, we have:

$$u_i = v_i^0 \oplus v_i^1 \oplus w$$

Why Consistency Check works?

We need to show that: $z' = \psi' \oplus (\Delta \cdot w')$

Proof:

The defination of the z, we have:

$$z_i = v_i^{\Delta_i} \oplus (\Delta_i \cdot u_i)$$

and the defination of the u_i, we have:

$$u_i = v_i^0 \oplus v_i^1 \oplus w$$

Writing u_i in the first equation:

$$z_i = v_i^{\Delta_i} \oplus \left[\Delta_i \cdot (v_i^0 \oplus v_i^1 \oplus w)\right]$$

Why Consistency Check works?

We need to show that: $z' = \psi' \oplus (\Delta \cdot w')$

Proof:

The defination of the z, we have:

$$z_i = v_i^{\Delta_i} \oplus (\Delta_i \cdot u_i)$$

and the defination of the u_i, we have:

$$u_i = v_i^0 \oplus v_i^1 \oplus w$$

Writing u_i in the first equation:

$$z_i = v_i^{\Delta_i} \oplus \left[\Delta_i \cdot (v_i^0 \oplus v_i^1 \oplus w)\right]$$

We have two cases

Why Consistency Check works?

We need to show that: $z' = \psi' \oplus (\Delta \cdot w')$
Proof:
The defination of the z, we have:

$$z_i = v_i^{\Delta_i} \oplus (\Delta_i \cdot u_i)$$

and the defination of the u_i, we have:

$$u_i = v_i^0 \oplus v_i^1 \oplus w$$

Writing u_i in the first equation:

$$z_i = v_i^{\Delta_i} \oplus \left[\Delta_i \cdot (v_i^0 \oplus v_i^1 \oplus w)\right]$$

We have two cases
Case 1: $\Delta_i = 0$

$$z_i = v_i^0 \oplus 0 = v_i^0$$

Why Consistency Check works?

We need to show that: $z' = \psi' \oplus (\Delta \cdot w')$
Proof:
The defination of the z, we have:

$$z_i = v_i^{\Delta_i} \oplus (\Delta_i \cdot u_i)$$

and the defination of the u_i, we have:

$$u_i = v_i^0 \oplus v_i^1 \oplus w$$

Writing u_i in the first equation:

$$z_i = v_i^{\Delta_i} \oplus \left[\Delta_i \cdot (v_i^0 \oplus v_i^1 \oplus w)\right]$$

We have two cases
Case 1: $\Delta_i = 0$

$$z_i = v_i^0 \oplus 0 = v_i^0$$

Case 2: $\Delta_i = 1$

$$z_i = v_i^1 \oplus (v_i^0 \oplus v_i^1 \oplus w) = v_i^0 \oplus w$$

Why Consistency Check works?

We need to show that: $z' = \psi' \oplus (\Delta \cdot w')$
Proof:
The defination of the z, we have:

$$z_i = v_i^{\Delta_i} \oplus (\Delta_i \cdot u_i)$$

and the defination of the u_i, we have:

$$u_i = v_i^0 \oplus v_i^1 \oplus w$$

Writing u_i in the first equation:

$$z_i = v_i^{\Delta_i} \oplus \left[\Delta_i \cdot (v_i^0 \oplus v_i^1 \oplus w)\right]$$

We have two cases
Case 1: $\Delta_i = 0$

$$z_i = v_i^0 \oplus 0 = v_i^0$$

Case 2: $\Delta_i = 1$

$$z_i = v_i^1 \oplus (v_i^0 \oplus v_i^1 \oplus w) = v_i^0 \oplus w$$

Which means:

$$z_i = \begin{cases} v_i^0 & \text{if } \Delta_i = 0 \\ v_i^0 \oplus w & \text{if } \Delta_i = 1 \end{cases}$$

Why Consistency Check works?

We need to show that: $z' = \psi' \oplus (\Delta \cdot w')$
Proof:
The defination of the z, we have:

$$z_i = v_i^{\Delta_i} \oplus (\Delta_i \cdot u_i)$$

and the defination of the u_i, we have:

$$u_i = v_i^0 \oplus v_i^1 \oplus w$$

Writing u_i in the first equation:

$$z_i = v_i^{\Delta_i} \oplus \left[\Delta_i \cdot (v_i^0 \oplus v_i^1 \oplus w)\right]$$

We have two cases
Case 1: $\Delta_i = 0$

$$z_i = v_i^0 \oplus 0 = v_i^0$$

Case 2: $\Delta_i = 1$

$$z_i = v_i^1 \oplus (v_i^0 \oplus v_i^1 \oplus w) = v_i^0 \oplus w$$

Which means:

$$z_i = \begin{cases} v_i^0 & \text{if } \Delta_i = 0 \\ v_i^0 \oplus w & \text{if } \Delta_i = 1 \end{cases}$$

So we can write $z_i = v_i^0 \oplus (\Delta_i \cdot w)$

Why Consistency Check works?

$$z_i = v_i^0 \oplus (\Delta_i \cdot w)$$

Why Consistency Check works?

$$z_i = v_i^0 \oplus (\Delta_i \cdot w)$$

When we transpose z_i, we have

$$\zeta_j = \psi_j \oplus (\Delta \cdot w_j)$$

Why Consistency Check works?

$$z_i = v_i^0 \oplus (\Delta_i \cdot w)$$

When we transpose z_i, we have

$$\zeta_j = \psi_j \oplus (\Delta \cdot w_j)$$

Multiplying by χ, we have

Why Consistency Check works?

$$z_i = v_i^0 \oplus (\Delta_i \cdot w)$$

When we transpose z_i, we have

$$\zeta_j = \psi_j \oplus (\Delta \cdot w_j)$$

Multiplying by χ, we have

$$z' = \bigoplus_j \zeta_j \cdot \chi_j$$

Why Consistency Check works?

$$z_i = v_i^0 \oplus (\Delta_i \cdot w)$$

When we transpose z_i, we have

$$\zeta_j = \psi_j \oplus (\Delta \cdot w_j)$$

Multiplying by χ, we have

$$\begin{aligned} z' &= \bigoplus_j \zeta_j \cdot \chi_j \\ &= \bigoplus_j (\psi_j \oplus (\Delta \cdot w_j)) \cdot \chi_j \end{aligned}$$

Why Consistency Check works?

$$z_i = v_i^0 \oplus (\Delta_i \cdot w)$$

When we transpose z_i, we have

$$\zeta_j = \psi_j \oplus (\Delta \cdot w_j)$$

Multiplying by χ, we have

$$\begin{aligned} z' &= \bigoplus_j \zeta_j \cdot \chi_j \\ &= \bigoplus_j (\psi_j \oplus (\Delta \cdot w_j)) \cdot \chi_j \\ &= \bigoplus_j (\psi_j \cdot \chi_j) \oplus (\Delta \cdot w_j \cdot \chi_j) \end{aligned}$$

Why Consistency Check works?

$$z_i = v_i^0 \oplus (\Delta_i \cdot w)$$

When we transpose z_i, we have

$$\zeta_j = \psi_j \oplus (\Delta \cdot w_j)$$

Multiplying by χ, we have

$$\begin{aligned} z' &= \bigoplus_j \zeta_j \cdot \chi_j \\ &= \bigoplus_j (\psi_j \oplus (\Delta \cdot w_j)) \cdot \chi_j \\ &= \bigoplus_j (\psi_j \cdot \chi_j) \oplus (\Delta \cdot w_j \cdot \chi_j) \\ &= \psi' \oplus (\Delta \cdot \bigoplus_j w_j \cdot \chi_j) \end{aligned}$$

Why Consistency Check works?

$$z_i = v_i^0 \oplus (\Delta_i \cdot w)$$

When we transpose z_i, we have

$$\zeta_j = \psi_j \oplus (\Delta \cdot w_j)$$

Multiplying by χ, we have

$$\begin{aligned}
z' &= \bigoplus_j \zeta_j \cdot \chi_j \\
&= \bigoplus_j (\psi_j \oplus (\Delta \cdot w_j)) \cdot \chi_j \\
&= \bigoplus_j (\psi_j \cdot \chi_j) \oplus (\Delta \cdot w_j \cdot \chi_j) \\
&= \psi' \oplus (\Delta \cdot \bigoplus_j w_j \cdot \chi_j) \\
&= \psi' \oplus (\Delta \cdot w')
\end{aligned}$$

Why KOS Works?

Alice calculates:

$$t_A := \{H(j \| \zeta_j)\}_{j \in [1,\ell']}$$
$$\tau := \{H(j \| (\zeta_j \oplus \Delta)) - t_{Aj} + \alpha_j\}_{j \in [1,\ell']}$$

Why KOS Works?

Alice calculates:

$$t_A := \{H(j \| \zeta_j)\}_{j \in [1, \ell']}$$
$$\tau := \{H(j \| (\zeta_j \oplus \Delta)) - t_{Aj} + \alpha_j\}_{j \in [1, \ell']}$$

Bob calculates:

$$t_B := \begin{cases} -H(j \| \psi_j) & \text{if } w_j = 0 \\ \tau_j - H(j \| \psi_j) & \text{if } w_j = 1 \end{cases}$$

Why KOS Works?

Alice calculates:

$$t_A := \{H(j \| \zeta_j)\}_{j \in [1, \ell']}$$
$$\tau := \{H(j \| (\zeta_j \oplus \Delta)) - t_{Aj} + \alpha_j\}_{j \in [1, \ell']}$$

Bob calculates:

$$t_B := \begin{cases} -H(j \| \psi_j) & \text{if } w_j = 0 \\ \tau_j - H(j \| \psi_j) & \text{if } w_j = 1 \end{cases}$$

Claim: $t_A + t_B = \omega \cdot \alpha$

Why KOS Works?

Alice calculates:

$$t_A := \{H(j \| \zeta_j)\}_{j \in [1,\ell']}$$
$$\tau := \{H(j \| (\zeta_j \oplus \Delta)) - t_{Aj} + \alpha_j\}_{j \in [1,\ell']}$$

Bob calculates:

$$t_B := \begin{cases} -H(j \| \psi_j) & \text{if } w_j = 0 \\ \tau_j - H(j \| \psi_j) & \text{if } w_j = 1 \end{cases}$$

Claim: $t_A + t_B = \omega \cdot \alpha$
Proof:

Why KOS Works?

Alice calculates:

$$t_A := \{H(j \| \zeta_j)\}_{j \in [1,\ell']}$$
$$\tau := \{H(j \| (\zeta_j \oplus \Delta)) - t_{Aj} + \alpha_j\}_{j \in [1,\ell']}$$

Bob calculates:

$$t_B := \begin{cases} -H(j \| \psi_j) & \text{if } w_j = 0 \\ \tau_j - H(j \| \psi_j) & \text{if } w_j = 1 \end{cases}$$

Claim: $t_A + t_B = \omega \cdot \alpha$
Proof: We know that $\zeta_j = \psi_j \oplus (\Delta \cdot w_j)$

Why KOS Works?

Alice calculates:

$$t_A := \{H(j\|\zeta_j)\}_{j\in[1,\ell']}$$
$$\tau := \{H(j\|(\zeta_j \oplus \Delta)) - t_{Aj} + \alpha_j\}_{j\in[1,\ell']}$$

Bob calculates:

$$t_B := \begin{cases} -H(j\|\psi_j) & \text{if } w_j = 0 \\ \tau_j - H(j\|\psi_j) & \text{if } w_j = 1 \end{cases}$$

Claim: $t_A + t_B = \omega \cdot \alpha$
Proof: We know that $\zeta_j = \psi_j \oplus (\Delta \cdot w_j)$
Case 1: $w_j = 0$

Why KOS Works?

Alice calculates:

$$t_A := \{H(j\|\zeta_j)\}_{j\in[1,\ell']}$$
$$\tau := \{H(j\|(\zeta_j \oplus \Delta)) - t_{Aj} + \alpha_j\}_{j\in[1,\ell']}$$

Bob calculates:

$$t_B := \begin{cases} -H(j\|\psi_j) & \text{if } w_j = 0 \\ \tau_j - H(j\|\psi_j) & \text{if } w_j = 1 \end{cases}$$

Claim: $t_A + t_B = \omega \cdot \alpha$
Proof: We know that $\zeta_j = \psi_j \oplus (\Delta \cdot w_j)$
Case 1: $w_j = 0$

$$\zeta_j = \psi_j \oplus 0 = \psi_j$$

Why KOS Works?

Alice calculates:

$$t_A := \{H(j \| \zeta_j)\}_{j \in [1,\ell']}$$
$$\tau := \{H(j \| (\zeta_j \oplus \Delta)) - t_{Aj} + \alpha_j\}_{j \in [1,\ell']}$$

Bob calculates:

$$t_B := \begin{cases} -H(j \| \psi_j) & \text{if } w_j = 0 \\ \tau_j - H(j \| \psi_j) & \text{if } w_j = 1 \end{cases}$$

Claim: $t_A + t_B = \omega \cdot \alpha$
Proof: We know that $\zeta_j = \psi_j \oplus (\Delta \cdot w_j)$
Case 1: $w_j = 0$

$$\zeta_j = \psi_j \oplus 0 = \psi_j$$
$$t_{Aj} = H(j \| \psi_j)$$

Why KOS Works?

Alice calculates:

$$t_A := \{H(j\|\zeta_j)\}_{j\in[1,\ell']}$$
$$\tau := \{H(j\|(\zeta_j \oplus \Delta)) - t_{Aj} + \alpha_j\}_{j\in[1,\ell']}$$

Bob calculates:

$$t_B := \begin{cases} -H(j\|\psi_j) & \text{if } w_j = 0 \\ \tau_j - H(j\|\psi_j) & \text{if } w_j = 1 \end{cases}$$

Claim: $t_A + t_B = \omega \cdot \alpha$
Proof: We know that $\zeta_j = \psi_j \oplus (\Delta \cdot w_j)$
Case 1: $w_j = 0$

$$\zeta_j = \psi_j \oplus 0 = \psi_j$$
$$t_{Aj} = H(j\|\psi_j)$$
$$\tau_j = H(j\|(\psi_j \oplus \Delta)) - t_{Aj} + \alpha_j$$

Why KOS Works?

Alice calculates:

$$t_A := \{H(j \| \zeta_j)\}_{j \in [1,\ell']}$$
$$\tau := \{H(j \| (\zeta_j \oplus \Delta)) - t_{Aj} + \alpha_j\}_{j \in [1,\ell']}$$

Bob calculates:

$$t_B := \begin{cases} -H(j \| \psi_j) & \text{if } w_j = 0 \\ \tau_j - H(j \| \psi_j) & \text{if } w_j = 1 \end{cases}$$

Claim: $t_A + t_B = \omega \cdot \alpha$
Proof: We know that $\zeta_j = \psi_j \oplus (\Delta \cdot w_j)$
Case 1: $w_j = 0$

$$\zeta_j = \psi_j \oplus 0 = \psi_j$$
$$t_{Aj} = H(j \| \psi_j)$$
$$\tau_j = H(j \| (\psi_j \oplus \Delta)) - t_{Aj} + \alpha_j$$

Calculation of Bob:

$$t_{Bj} = -H(j \| \psi_j)$$

Why KOS Works?

Alice calculates:

$$t_A := \{H(j\|\zeta_j)\}_{j\in[1,\ell']}$$
$$\tau := \{H(j\|(\zeta_j \oplus \Delta)) - t_{Aj} + \alpha_j\}_{j\in[1,\ell']}$$

Bob calculates:

$$t_B := \begin{cases} -H(j\|\psi_j) & \text{if } w_j = 0 \\ \tau_j - H(j\|\psi_j) & \text{if } w_j = 1 \end{cases}$$

Claim: $t_A + t_B = \omega \cdot \alpha$
Proof: We know that $\zeta_j = \psi_j \oplus (\Delta \cdot w_j)$
Case 1: $w_j = 0$

$$\begin{aligned} \zeta_j &= \psi_j \oplus 0 = \psi_j \\ t_{Aj} &= H(j\|\psi_j) \\ \tau_j &= H(j\|(\psi_j \oplus \Delta)) - t_{Aj} + \alpha_j \end{aligned}$$

Calculation of Bob:

$$t_{Bj} = -H(j\|\psi_j)$$

Summing two shares:

$$\begin{aligned} t_{Aj} + t_{Bj} &= H(j\|\psi_j) - H(j\|\psi_j) \\ &= 0 \\ &= \alpha_j \cdot w_j \quad (\text{ since } w_j = 0) \end{aligned}$$

Why KOS Works?

Case 2: $w_j = 1$

$$\begin{aligned} \zeta_j &= \psi_j \oplus \Delta \\ t_{Aj} &= H(j \| (\psi_j \oplus \Delta)) \\ \tau_j &= H(j \| \psi_j) - t_{Aj} + \alpha_j \\ &= H(j \| \psi_j) - H(j \| (\psi_j \oplus \Delta)) + \alpha_j \end{aligned}$$

Why KOS Works?

Case 2: $w_j = 1$

$$\begin{aligned} \zeta_j &= \psi_j \oplus \Delta \\ t_{Aj} &= H(j \| (\psi_j \oplus \Delta)) \\ \tau_j &= H(j \| \psi_j) - t_{Aj} + \alpha_j \\ &= H(j \| \psi_j) - H(j \| (\psi_j \oplus \Delta)) + \alpha_j \end{aligned}$$

Calculation of Bob::

$$\begin{aligned} t_{Bj} &= \tau_j - H(j \| \psi_j) \\ &= (H(j \| \psi_j) - t_{Aj} + \alpha_j) - H(j \| \psi_j) \\ &= -t_{Aj} + \alpha_j \end{aligned}$$

Why KOS Works?

Case 2: $w_j = 1$

$$\begin{aligned}
\zeta_j &= \psi_j \oplus \Delta \\
t_{Aj} &= H(j \| (\psi_j \oplus \Delta)) \\
\tau_j &= H(j \| \psi_j) - t_{Aj} + \alpha_j \\
&= H(j \| \psi_j) - H(j \| (\psi_j \oplus \Delta)) + \alpha_j
\end{aligned}$$

Calculation of Bob::

$$\begin{aligned}
t_{Bj} &= \tau_j - H(j \| \psi_j) \\
&= (H(j \| \psi_j) - t_{Aj} + \alpha_j) - H(j \| \psi_j) \\
&= -t_{Aj} + \alpha_j
\end{aligned}$$

Summing t_A and t_B, we have:

$$\begin{aligned}
t_{Aj} + t_{Bj} &= t_{Aj} - t_{Aj} + \alpha_j \\
&= \alpha_j \\
&= \alpha_j \cdot w_j \quad (\text{ since } w_j = 1)
\end{aligned}$$

Why KOS Works?

Case 2: $w_j = 1$

$$\begin{aligned}\zeta_j &= \psi_j \oplus \Delta \\ t_{Aj} &= H(j \| (\psi_j \oplus \Delta)) \\ \tau_j &= H(j \| \psi_j) - t_{Aj} + \alpha_j \\ &= H(j \| \psi_j) - H(j \| (\psi_j \oplus \Delta)) + \alpha_j\end{aligned}$$

Calculation of Bob::

$$\begin{aligned}t_{Bj} &= \tau_j - H(j \| \psi_j) \\ &= (H(j \| \psi_j) - t_{Aj} + \alpha_j) - H(j \| \psi_j) \\ &= -t_{Aj} + \alpha_j\end{aligned}$$

Summing t_A and t_B, we have:

$$\begin{aligned}t_{Aj} + t_{Bj} &= t_{Aj} - t_{Aj} + \alpha_j \\ &= \alpha_j \\ &= \alpha_j \cdot w_j \quad (\text{ since } w_j = 1)\end{aligned}$$

In both cases

$$t_{Aj} + t_{Bj} = \alpha_j \cdot w_j$$

$\pi^{\ell}_{\mathsf{2PMul}}$ - Two-party Multiplication

Alice		Bob
$\tilde{\mathbf{a}} \leftarrow \mathbb{Z}_q^{\ell}$ and $\hat{\mathbf{a}} \leftarrow \mathbb{Z}_q^{\ell}$		$\boldsymbol{\beta} \leftarrow \{0,1\}^{\eta}$
$\boldsymbol{\alpha} := \{\tilde{\mathbf{a}}_1 \Vert \hat{\mathbf{a}}_1\}_{j\in[\xi]} \parallel \ldots \parallel \{\tilde{\mathbf{a}}_{\ell} \Vert \hat{\mathbf{a}}_{\ell}\}_{j\in[\xi]}$		$\tilde{\mathbf{b}} := \left\{\left\langle \mathbf{g}, \{\boldsymbol{\beta}_{i\cdot\xi+j}\}_{j\in[\xi]}\right\rangle\right\}_{i\in[0,\ell)}$
$\boldsymbol{\alpha} \longrightarrow$ $\boldsymbol{\omega}_A \longleftarrow$	$\boxed{\mathcal{F}^{\eta}_{\mathrm{COTe}}}$	$\longleftarrow \boldsymbol{\beta}$ $\longrightarrow \boldsymbol{\omega}_B$
$\{\tilde{\mathbf{z}}_{A,j} \Vert \hat{\mathbf{z}}_{A,j}\}_{j\in[\eta]} = \boldsymbol{\omega}_A$		$\{\tilde{\mathbf{z}}_{\mathsf{B},j} \Vert \hat{\mathbf{z}}_{B,j}\}_{j\in[\eta]} = \boldsymbol{\omega}_B$
$\tilde{\chi} \leftarrow H^{\ell}(1\Vert\text{transcript})$	(Both calculate $\tilde{\chi}$ and $\hat{\chi}$)	$\hat{\chi} \leftarrow H^{\ell}(2\Vert\text{transcript})$

$$\mathbf{r} := \left\{ \sum_{i\in[0,\ell)} \tilde{\chi}_{i+1} \cdot \tilde{\mathbf{z}}_{\mathsf{A},i\cdot\xi+j} + \hat{\chi}_{i+1} \cdot \hat{\mathbf{z}}_{\mathsf{A},i\cdot\xi+j} \right\}_{j\in[\xi]}$$

$$\mathbf{u} := \{\tilde{\chi}_i \cdot \tilde{\mathbf{a}}_i + \hat{\chi}_i \cdot \hat{\mathbf{a}}_i\}_{i\in[\ell]} \qquad \xrightarrow{\mathbf{u},\mathbf{r}}$$

$$\forall_{j\in[\xi]} \left(\mathbf{r}_j + \sum_{i\in[0,\ell)} \tilde{\chi}_{i+1} \cdot \tilde{\mathbf{z}}_{\mathsf{B},i\cdot\xi+j} + \hat{\chi}_{i+1} \cdot \hat{\mathbf{z}}_{\mathsf{B},i\cdot\xi+j} \stackrel{?}{=} \sum_{i\in[0,\ell)} \boldsymbol{\beta}_{i\cdot\xi+j} \cdot \mathbf{u}_{i+1} \right)$$

$$\gamma_{\mathsf{A}} := \{\mathbf{a}_i - \tilde{\mathbf{a}}_i\}_{i\in[\ell]} \qquad \gamma_{\mathsf{B}} := \left\{\mathbf{b}_i - \tilde{\mathbf{b}}_i\right\}_{i\in[\ell]}$$

$$\xrightarrow{\gamma_{\mathsf{A}}} \qquad \xleftarrow{\gamma_{\mathsf{B}}}$$

$$\mathbf{z}_{\mathsf{A}} := \left\{\mathbf{a}_{i+1} \cdot \gamma_{\mathsf{B},i+1} + \textstyle\sum_{j\in[\xi]} \mathbf{g}_j \cdot \tilde{\mathbf{z}}_{\mathsf{A},i,\xi+j}\right\}_{i\in[0,\ell)} \qquad \mathbf{z}_{\mathsf{B}} := \left\{\tilde{\mathbf{b}}_{i+1} \cdot \gamma_{\mathsf{A},i+1} + \textstyle\sum_{j\in[\xi]} \mathbf{g}_j \cdot \tilde{\mathbf{z}}_{\mathsf{B},i,\xi+j}\right\}_{i\in[0,\ell)}$$

$$\mathbf{z}_{\mathsf{A}} + \mathbf{z}_{\mathsf{B}} = \mathbf{a} \cdot \mathbf{b}$$

Why MtA Works?

$$z_A = \left\{ a_i \cdot \gamma_{B,i} + \sum_j g_j \cdot \tilde{z}_{A,i-\xi+j} \right\}_{i \in [\ell]} \qquad z_B = \left\{ \tilde{b}_i \cdot \gamma_{A,i} + \sum_j g_j \cdot \tilde{z}_{B,i-\xi+j} \right\}_{i \in [\ell]}$$

Why MtA Works?

$$z_A = \left\{ a_i \cdot \gamma_{B,i} + \sum_j g_j \cdot \tilde{z}_{A,i-\xi+j} \right\}_{i\in[\ell]} \qquad z_B = \left\{ \tilde{b}_i \cdot \gamma_{A,i} + \sum_j g_j \cdot \tilde{z}_{B,i-\xi+j} \right\}_{i\in[\ell]}$$

Summing these values,

$$z_A + z_B = a \cdot \gamma_B + \tilde{b} \cdot \gamma_A + \sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B)_{i-\xi+j}$$

Why MtA Works?

$$z_A = \left\{ a_i \cdot \gamma_{B,i} + \sum_j g_j \cdot \tilde{z}_{A,i-\xi+j} \right\}_{i\in[\ell]} \qquad z_B = \left\{ \tilde{b}_i \cdot \gamma_{A,i} + \sum_j g_j \cdot \tilde{z}_{B,i-\xi+j} \right\}_{i\in[\ell]}$$

Summing these values,

$$z_A + z_B = a \cdot \gamma_B + \tilde{b} \cdot \gamma_A + \sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B)_{i-\xi+j}$$

Since $\gamma_A = a - \tilde{a}$, $\gamma_B = b - \tilde{b}$ we have:

Why MtA Works?

$$z_A = \left\{ a_i \cdot \gamma_{B,i} + \sum_j g_j \cdot \tilde{z}_{A,i-\xi+j} \right\}_{i\in[\ell]} \qquad z_B = \left\{ \tilde{b}_i \cdot \gamma_{A,i} + \sum_j g_j \cdot \tilde{z}_{B,i-\xi+j} \right\}_{i\in[\ell]}$$

Summing these values,

$$z_A + z_B = a \cdot \gamma_B + \tilde{b} \cdot \gamma_A + \sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B)_{i-\xi+j}$$

Since $\gamma_A = a - \tilde{a}$, $\gamma_B = b - \tilde{b}$ we have:

$$z_A + z_B = a \cdot (b - \tilde{b}) + \tilde{b} \cdot (a - \tilde{a}) + \sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B)$$

Why MtA Works?

$$z_A = \left\{ a_i \cdot \gamma_{B,i} + \sum_j g_j \cdot \tilde{z}_{A,i-\xi+j} \right\}_{i \in [\ell]} \qquad z_B = \left\{ \tilde{b}_i \cdot \gamma_{A,i} + \sum_j g_j \cdot \tilde{z}_{B,i-\xi+j} \right\}_{i \in [\ell]}$$

Summing these values,

$$z_A + z_B = a \cdot \gamma_B + \tilde{b} \cdot \gamma_A + \sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B)_{i-\xi+j}$$

Since $\gamma_A = a - \tilde{a}$, $\gamma_B = b - \tilde{b}$ we have:

$$\begin{aligned} z_A + z_B &= a \cdot (b - \tilde{b}) + \tilde{b} \cdot (a - \tilde{a}) + \sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B) \\ &= a \cdot b - a \cdot \tilde{b} + a \cdot \tilde{b} - \tilde{a} \cdot \tilde{b} + \sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B) \end{aligned}$$

Why MtA Works?

$$z_A = \left\{ a_i \cdot \gamma_{B,i} + \sum_j g_j \cdot \tilde{z}_{A,i-\xi+j} \right\}_{i \in [\ell]} \qquad z_B = \left\{ \tilde{b}_i \cdot \gamma_{A,i} + \sum_j g_j \cdot \tilde{z}_{B,i-\xi+j} \right\}_{i \in [\ell]}$$

Summing these values,

$$z_A + z_B = a \cdot \gamma_B + \tilde{b} \cdot \gamma_A + \sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B)_{i-\xi+j}$$

Since $\gamma_A = a - \tilde{a}$, $\gamma_B = b - \tilde{b}$ we have:

$$\begin{aligned} z_A + z_B &= a \cdot (b - \tilde{b}) + \tilde{b} \cdot (a - \tilde{a}) + \sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B) \\ &= a \cdot b - a \cdot \tilde{b} + a \cdot \tilde{b} - \tilde{a} \cdot \tilde{b} + \sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B) \\ &= a \cdot b - \tilde{a} \cdot \tilde{b} + \sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B) \end{aligned}$$

$$= a \cdot b - \tilde{a} \cdot \tilde{b} + \sum_{j} g_j \cdot (\tilde{z}_A + \tilde{z}_B)$$

$$= a \cdot b - \tilde{a} \cdot \tilde{b} + \sum_{j} g_j \cdot (\tilde{z}_A + \tilde{z}_B)$$

By the property of COT, we have $\tilde{z}_A + \tilde{z}_B = \tilde{a} \cdot \beta$

$$= a \cdot b - \tilde{a} \cdot \tilde{b} + \sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B)$$

By the property of COT, we have $\tilde{z}_A + \tilde{z}_B = \tilde{a} \cdot \beta$

$$\sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B) = \sum_j g_j \cdot \tilde{a} \cdot \beta_j$$

$$= a \cdot b - \tilde{a} \cdot \tilde{b} + \sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B)$$

By the property of COT, we have $\tilde{z}_A + \tilde{z}_B = \tilde{a} \cdot \beta$

$$\begin{aligned} \sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B) &= \sum_j g_j \cdot \tilde{a} \cdot \beta_j \\ &= \tilde{a} \cdot \sum_j g_j \beta_j \end{aligned}$$

$$= a \cdot b - \tilde{a} \cdot \tilde{b} + \sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B)$$

By the property of COT, we have $\tilde{z}_A + \tilde{z}_B = \tilde{a} \cdot \beta$

$$\begin{aligned}
\sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B) &= \sum_j g_j \cdot \tilde{a} \cdot \beta_j \\
&= \tilde{a} \cdot \sum_j g_j \beta_j \\
&= \tilde{a} \cdot \tilde{b} \quad (\text{since } \tilde{b} = \langle g, \beta \rangle)
\end{aligned}$$

$$= a \cdot b - \tilde{a} \cdot \tilde{b} + \sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B)$$

By the property of COT, we have $\tilde{z}_A + \tilde{z}_B = \tilde{a} \cdot \beta$

$$\begin{aligned}
\sum_j g_j \cdot (\tilde{z}_A + \tilde{z}_B) &= \sum_j g_j \cdot \tilde{a} \cdot \beta_j \\
&= \tilde{a} \cdot \sum_j g_j \beta_j \\
&= \tilde{a} \cdot \tilde{b} \quad (\text{since } \tilde{b} = \langle g, \beta \rangle)
\end{aligned}$$

Hence,

$$z_A + z_B = a \cdot b - \tilde{a} \cdot \tilde{b} + \tilde{a} \cdot \tilde{b} = a \cdot b$$

Why Consistency Check Works?

Alice calculates:

$$r = \sum_i \tilde{x}_i \cdot \tilde{z}_{A,i} + \hat{x}_i \cdot \hat{z}_{A,i}, \quad u = \tilde{x} \cdot \tilde{a} + \hat{x} \cdot \hat{a}$$

Why Consistency Check Works?

Alice calculates:

$$r = \sum_i \tilde{x}_i \cdot \tilde{z}_{A,i} + \hat{x}_i \cdot \hat{z}_{A,i}, \quad u = \tilde{x} \cdot \tilde{a} + \hat{x} \cdot \hat{a}$$

Bob checks that:

$$r + \sum_i \tilde{x}_i \cdot \tilde{z}_{B,i} + \hat{x}_i \cdot \hat{z}_{B,i} \stackrel{?}{=} \sum_i \beta_i \cdot u_i$$

Why Consistency Check Works?

Alice calculates:

$$r = \sum_i \tilde{x}_i \cdot \tilde{z}_{A,i} + \hat{x}_i \cdot \hat{z}_{A,i}, \quad u = \tilde{x} \cdot \tilde{a} + \hat{x} \cdot \hat{a}$$

Bob checks that:

$$r + \sum_i \tilde{x}_i \cdot \tilde{z}_{B,i} + \hat{x}_i \cdot \hat{z}_{B,i} \stackrel{?}{=} \sum_i \beta_i \cdot u_i$$

By COT $\tilde{z}_{A,i}$, $\hat{z}_B$ ve $\tilde{z}_{A,i}$, $\hat{z}_B$ satisfy

$$\tilde{z}_A + \tilde{z}_B = \tilde{a} \cdot \beta, \quad \hat{z}_A + \hat{z}_B = \hat{a} \cdot \beta$$

Why Consistency Check Works?

Alice calculates:

$$r = \sum_i \tilde{x}_i \cdot \tilde{z}_{A,i} + \hat{x}_i \cdot \hat{z}_{A,i}, \quad u = \tilde{x} \cdot \tilde{a} + \hat{x} \cdot \hat{a}$$

Bob checks that:

$$r + \sum_i \tilde{x}_i \cdot \tilde{z}_{B,i} + \hat{x}_i \cdot \hat{z}_{B,i} \stackrel{?}{=} \sum_i \beta_i \cdot u_i$$

By COT $\tilde{z}_{A,i}$, $\hat{z}_B$ ve $\tilde{z}_{A,i}$, $\hat{z}_B$ satisfy

$$\tilde{z}_A + \tilde{z}_B = \tilde{a} \cdot \beta, \quad \hat{z}_A + \hat{z}_B = \hat{a} \cdot \beta$$

LHS of Bob's equation:

Why Consistency Check Works?

Alice calculates:

$$r = \sum_i \tilde{x}_i \cdot \tilde{z}_{A,i} + \hat{x}_i \cdot \hat{z}_{A,i}, \quad u = \tilde{x} \cdot \tilde{a} + \hat{x} \cdot \hat{a}$$

Bob checks that:

$$r + \sum_i \tilde{x}_i \cdot \tilde{z}_{B,i} + \hat{x}_i \cdot \hat{z}_{B,i} \stackrel{?}{=} \sum_i \beta_i \cdot u_i$$

By COT $\tilde{z}_{A,i}$, $\hat{z}_B$ ve $\tilde{z}_{A,i}$, $\hat{z}_B$ satisfy

$$\tilde{z}_A + \tilde{z}_B = \tilde{a} \cdot \beta, \quad \hat{z}_A + \hat{z}_B = \hat{a} \cdot \beta$$

LHS of Bob's equation:

$$r + \sum_i (\tilde{x}_i \cdot \tilde{z}_{B,i} + \hat{x}_i \cdot \hat{z}_{B,i}) = \sum_i \tilde{x}_i \cdot (\tilde{z}_{A,i} + \tilde{z}_{B,i}) + \hat{x}_i \cdot (\hat{z}_{A,i} + \hat{z}_{B,i})$$

Why Consistency Check Works?

Alice calculates:

$$r = \sum_i \tilde{x}_i \cdot \tilde{z}_{A,i} + \hat{x}_i \cdot \hat{z}_{A,i}, \quad u = \tilde{x} \cdot \tilde{a} + \hat{x} \cdot \hat{a}$$

Bob checks that:

$$r + \sum_i \tilde{x}_i \cdot \tilde{z}_{B,i} + \hat{x}_i \cdot \hat{z}_{B,i} \stackrel{?}{=} \sum_i \beta_i \cdot u_i$$

By COT $\tilde{z}_{A,i}$, $\hat{z}_B$ ve $\tilde{z}_{A,i}$, $\hat{z}_B$ satisfy

$$\tilde{z}_A + \tilde{z}_B = \tilde{a} \cdot \beta, \quad \hat{z}_A + \hat{z}_B = \hat{a} \cdot \beta$$

LHS of Bob's equation:

$$\begin{aligned} r + \sum_i (\tilde{x}_i \cdot \tilde{z}_{B,i} + \hat{x}_i \cdot \hat{z}_{B,i}) &= \sum_i \tilde{x}_i \cdot (\tilde{z}_{A,i} + \tilde{z}_{B,i}) + \hat{x}_i \cdot (\hat{z}_{A,i} + \hat{z}_{B,i}) \\ &= \sum_i \tilde{x}_i \cdot \tilde{a}_i \cdot \beta_i + \hat{x}_i \cdot \hat{a}_i \cdot \beta_i \end{aligned}$$

Why Consistency Check Works?

Alice calculates:

$$r = \sum_i \tilde{x}_i \cdot \tilde{z}_{A,i} + \hat{x}_i \cdot \hat{z}_{A,i}, \quad u = \tilde{x} \cdot \tilde{a} + \hat{x} \cdot \hat{a}$$

Bob checks that:

$$r + \sum_i \tilde{x}_i \cdot \tilde{z}_{B,i} + \hat{x}_i \cdot \hat{z}_{B,i} \stackrel{?}{=} \sum_i \beta_i \cdot u_i$$

By COT $\tilde{z}_{A,i}$, $\hat{z}_B$ ve $\tilde{z}_{A,i}$, $\hat{z}_B$ satisfy

$$\tilde{z}_A + \tilde{z}_B = \tilde{a} \cdot \beta, \quad \hat{z}_A + \hat{z}_B = \hat{a} \cdot \beta$$

LHS of Bob's equation:

$$\begin{aligned} r + \sum_i (\tilde{x}_i \cdot \tilde{z}_{B,i} + \hat{x}_i \cdot \hat{z}_{B,i}) &= \sum_i \tilde{x}_i \cdot (\tilde{z}_{A,i} + \tilde{z}_{B,i}) + \hat{x}_i \cdot (\hat{z}_{A,i} + \hat{z}_{B,i}) \\ &= \sum_i \tilde{x}_i \cdot \tilde{a}_i \cdot \beta_i + \hat{x}_i \cdot \hat{a}_i \cdot \beta_i \\ &= \sum_i \beta_i \cdot (\tilde{x}_i \cdot \tilde{a}_i + \hat{x}_i \cdot \hat{a}_i) \end{aligned}$$

Why Consistency Check Works?

Alice calculates:

$$r = \sum_i \tilde{x}_i \cdot \tilde{z}_{A,i} + \hat{x}_i \cdot \hat{z}_{A,i}, \quad u = \tilde{x} \cdot \tilde{a} + \hat{x} \cdot \hat{a}$$

Bob checks that:

$$r + \sum_i \tilde{x}_i \cdot \tilde{z}_{B,i} + \hat{x}_i \cdot \hat{z}_{B,i} \stackrel{?}{=} \sum_i \beta_i \cdot u_i$$

By COT $\tilde{z}_{A,i}$, $\hat{z}_B$ ve $\tilde{z}_{A,i}$, $\hat{z}_B$ satisfy

$$\tilde{z}_A + \tilde{z}_B = \tilde{a} \cdot \beta, \quad \hat{z}_A + \hat{z}_B = \hat{a} \cdot \beta$$

LHS of Bob's equation:

$$\begin{aligned} r + \sum_i (\tilde{x}_i \cdot \tilde{z}_{B,i} + \hat{x}_i \cdot \hat{z}_{B,i}) &= \sum_i \tilde{x}_i \cdot (\tilde{z}_{A,i} + \tilde{z}_{B,i}) + \hat{x}_i \cdot (\hat{z}_{A,i} + \hat{z}_{B,i}) \\ &= \sum_i \tilde{x}_i \cdot \tilde{a}_i \cdot \beta_i + \hat{x}_i \cdot \hat{a}_i \cdot \beta_i \\ &= \sum_i \beta_i \cdot (\tilde{x}_i \cdot \tilde{a}_i + \hat{x}_i \cdot \hat{a}_i) \\ &= \sum_i \beta_i \cdot u_i \end{aligned}$$

t-party Modular Inverse Sampling

- Each party P_i for $i \in P$ samples $k_i \leftarrow \mathbb{Z}_q$ and a pad $\phi_i \leftarrow \mathbb{Z}_q$, set these values as $\psi_i^1 := \{k_i, \frac{\phi_i}{k_i}\}$ and commit the pad ϕ_i

t-party Modular Inverse Sampling

- Each party P_i for $i \in P$ samples $k_i \leftarrow \mathbb{Z}_q$ and a pad $\phi_i \leftarrow \mathbb{Z}_q$, set these values as $\psi_i^1 := \{k_i, \frac{\phi_i}{k_i}\}$ and commit the pad ϕ_i
- For $\rho \in \lceil \log_2 t \rceil$:

t-party Modular Inverse Sampling

- Each party P_i for $i \in P$ samples $k_i \leftarrow \mathbb{Z}_q$ and a pad $\phi_i \leftarrow \mathbb{Z}_q$, set these values as $\psi_i^1 := \{k_i, \frac{\phi_i}{k_i}\}$ and commit the pad ϕ_i
- For $\rho \in \lceil \log_2 t \rceil$:
  - For each pair of parties P_i, P_j in each contiguous non-overlapping subgroup of 2^ρ parties from $\mathcal{P}$, if P_i and P_j have not previously interacted during the course of this invocation of $\pi_{\text{inv}}^{n,t}$, then they send

$$(\psi_i^\rho) \quad \text{and} \quad (\psi_j^\rho)$$

    to $\mathcal{F}_{\text{2PMul}}^{\ell}$, respectively, and receive

$$(\zeta_i^{\rho,j}) \quad \text{and} \quad (\zeta_j^{\rho,i}).$$

t-party Modular Inverse Sampling

- Each party P_i for $i \in P$ samples $k_i \leftarrow \mathbb{Z}_q$ and a pad $\phi_i \leftarrow \mathbb{Z}_q$, set these values as $\psi_i^1 := \{k_i, \frac{\phi_i}{k_i}\}$ and commit the pad ϕ_i
- For $\rho \in \lceil \log_2 t \rceil$:
  - For each pair of parties P_i, P_j in each contiguous non-overlapping subgroup of 2^ρ parties from $\mathcal{P}$, if P_i and P_j have not previously interacted during the course of this invocation of $\pi_{\mathrm{inv}}^{n,t}$, then they send

    $$(\psi_i^\rho) \quad \text{and} \quad (\psi_j^\rho)$$

    to $\mathcal{F}_{\mathrm{2PMul}}^{\ell}$, respectively, and receive

    $$(\zeta_i^{\rho,j}) \quad \text{and} \quad (\zeta_j^{\rho,i}).$$

  - Each party P_i privately computes $\psi_i^{\rho+1}$ to be the element-wise sum of its output shares for round ρ:

    $$\psi_i^{\rho+1} := \left\{ \sum_{j \in P^{\rho,i}} \zeta_{i,\ell}^{\rho,j} \right\}_{\ell \in [2]}$$

    where $P^{\rho,i} \subset \mathcal{P}$ is the subgroup with whom P_i interacted in round ρ such that $|P^{\rho,i}| = 2^{\rho-1}$.

Step-2 (4 Party Example)

$\rho = 1$: All parties come together two by two.

(P_1, P_2)

$$\psi_1^1 \longrightarrow \boxed{\mathcal{F}_{2PMU}} \longleftarrow \psi_2^1$$
$$\xi_1^{1,2} \longleftarrow \boxed{\mathcal{F}_{2PMU}} \longrightarrow \xi_2^{1,1}$$

P_1 $\quad$ P_2

$\psi_1^2 := \xi_1^{1,2}$ $\qquad$ $\psi_2^2 := \xi_2^{1,1}$

(P_3, P_4)

$$\psi_3^1 \longrightarrow \boxed{\mathcal{F}_{2PMU}} \longleftarrow \psi_4^1$$
$$\xi_3^{1,4} \longleftarrow \boxed{\mathcal{F}_{2PMU}} \longrightarrow \xi_4^{1,3}$$

P_3 $\quad$ P_4

$\psi_3^2 := \xi_3^{1,4}$ $\qquad$ $\psi^2 := \xi_4^{1,3}$

$\rho = 2$: All parties come together four by four.

(P_1, P_2, P_3, P_4)

(P_1, P_3)

$$\psi_1^2 \longrightarrow \boxed{\mathcal{F}_{2PMU}} \longleftarrow \psi_3^2$$
$$\xi_1^{2,3} \longleftarrow \boxed{\mathcal{F}_{2PMU}} \longrightarrow \xi_3^{2,1}$$

(P_1, P_4)

$$\psi_1^2 \longrightarrow \boxed{\mathcal{F}_{2PMU}} \longleftarrow \psi_4^2$$
$$\xi_1^{2,4} \longleftarrow \boxed{\mathcal{F}_{2PMU}} \longrightarrow \xi_4^{2,1}$$

(P_2, P_3)

$$\psi_2^2 \longrightarrow \boxed{\mathcal{F}_{2PMU}} \longleftarrow \psi_3^2$$
$$\xi_2^{2,3} \longleftarrow \boxed{\mathcal{F}_{2PMU}} \longrightarrow \xi_3^{2,2}$$

(P_2, P_4)

$$\psi_2^2 \longrightarrow \boxed{\mathcal{F}_{2PMU}} \longleftarrow \psi_4^2$$
$$\xi_2^{2,4} \longleftarrow \boxed{\mathcal{F}_{2PMU}} \longrightarrow \xi_4^{2,2}$$

P_1:

$\psi_1^3 := \xi_1^{2,3} + \xi_1^{2,4}$

P_2:

$\psi_2^3 := \xi_2^{2,3} + \xi_2^{2,4}$

P_3:

$\psi_3^2 := \xi_3^{2,1} + \xi_3^{2,2}$

P_4:

$\psi_4^3 := \xi_4^{2,1} + \xi_4^{2,2}$

$$\psi_1^1 \cdot \psi_2^1 \cdot \psi_3^1 \cdot \psi_4^1$$

$$(\psi_1^2 + \psi_2^2) \cdot (\psi_3^2 + \psi_4^2)$$

$$\psi_1^2 \cdot \psi_3^2 + \psi_1^2 \cdot \psi_4^2 + \psi_2^2 \cdot \psi_3^2 + \psi_2^2 \cdot \psi_4^2$$

$$\xi_1^{2,3} + \xi_3^{2,1} + \xi_1^{2,4} + \xi_4^{2,1} + \xi_2^{2,3} + \xi_3^{2,2} + \xi_2^{2,4} + \xi_4^{2,2}$$

$$\psi_1^3 + \psi_3^3 + \psi_2^3 + \psi_4^3$$

10 Party Example

2-by-2 group

(P_1, P_2) (P_3, P_4) (P_5, P_6) (P_7, P_8) (P_9, P_{10})

4-by 4 group

(P_1, P_2, P_3, P_4) (P_5, P_6, P_7, P_8) (P_9, P_{10})

(P_1, P_3) (P_1, P_4) (P_5, P_7) (P_5, P_8)

(P_2, P_3) (P_2, P_4) (P_6, P_7) (P_6, P_8)

8-by-8 group

$(P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8)$ (P_9, P_{10})

(P_1, P_5), (P_1, P_6), (P_1, P_7) (P_1, P_8)

(P_2, P_5), (P_2, P_6), (P_2, P_7), (P_2, P_8)

(P_3, P_5), (P_3, P_6), (P_3, P_7), (P_3, P_8)

(P_4, P_5), (P_4, P_6), (P_4, P_7), (P_4, P_8)

10-by-10

$(P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10})$

(P_1, P_9), (P_1, P_{10})

(P_2, P_9), (P_2, P_{10})

(P_3, P_9), (P_3, P_{10})

(P_4, P_9), (P_4, P_{10})

(P_5, P_9), (P_5, P_{10})

(P_6, P_9), (P_6, P_{10})

(P_7, P_9), (P_7, P_{10})

(P_8, P_9), (P_8, P_{10})

t-party Modular Inverse Sampling

- Each party P_i sets $\{u_i, \tilde{v}_i\} = \psi_i^{\lceil \log_2(t) \rceil + 1}$ and so

$$\sum_{i \in \mathcal{P}} u_i = \prod_{i \in \mathcal{P}} k_i \quad \text{and} \quad \sum_{i \in \mathcal{P}} \tilde{v}_i = \prod_{i \in \mathcal{P}} \frac{\phi_i}{k_i}.$$

t-party Modular Inverse Sampling

- Each party P_i sets $\{u_i, \tilde{v}_i\} = \psi_i^{\lceil \log_2(t) \rceil + 1}$ and so

$$\sum_{i \in \mathcal{P}} u_i = \prod_{i \in \mathcal{P}} k_i \qquad \text{and} \qquad \sum_{i \in \mathcal{P}} \tilde{v}_i = \prod_{i \in \mathcal{P}} \frac{\phi_i}{k_i}.$$

Consistency Check:

- Each party P_i sets $\{u_i, \tilde{v}_i\} = \psi_i^{\lceil \log_2(t) \rceil + 1}$ and so

$$\sum_{i \in \mathcal{P}} u_i = \prod_{i \in \mathcal{P}} k_i \quad \text{and} \quad \sum_{i \in \mathcal{P}} \tilde{v}_i = \prod_{i \in \mathcal{P}} \frac{\phi_i}{k_i}.$$

Consistency Check:

- Each party P_i computes $R_i := u_i \cdot G$ and commits R_i

t-party Modular Inverse Sampling

- Each party P_i sets $\{u_i, \tilde{v}_i\} = \psi_i^{\lceil \log_2(t) \rceil + 1}$ and so

$$\sum_{i \in \mathcal{P}} u_i = \prod_{i \in \mathcal{P}} k_i \quad \text{and} \quad \sum_{i \in \mathcal{P}} \tilde{v}_i = \prod_{i \in \mathcal{P}} \frac{\phi_i}{k_i}.$$

Consistency Check:

- Each party P_i computes $R_i := u_i \cdot G$ and commits R_i
- Upon being notified of all other parties' commitments, each party P_j open commitment of R_j. Then each party computes

$$R := \sum_{j \in \mathcal{P}} R_j.$$

t-party Modular Inverse Sampling

- Each party P_i sets $\{u_i, \tilde{v}_i\} = \psi_i^{\lceil \log_2(t) \rceil + 1}$ and so

$$\sum_{i \in \mathcal{P}} u_i = \prod_{i \in \mathcal{P}} k_i \quad \text{and} \quad \sum_{i \in \mathcal{P}} \tilde{v}_i = \prod_{i \in \mathcal{P}} \frac{\phi_i}{k_i}.$$

Consistency Check:

- Each party P_i computes $R_i := u_i \cdot G$ and commits R_i
- Upon being notified of all other parties' commitments, each party P_j open commitment of R_j. Then each party computes

$$R := \sum_{j \in \mathcal{P}} R_j.$$

- Each party calculates $\Gamma_i^1 := \tilde{v}_i \cdot R$ and commit Γ_i^1.

t-party Modular Inverse Sampling

- Each party P_i sets $\{u_i, \tilde{v}_i\} = \psi_i^{\lceil \log_2(t) \rceil + 1}$ and so

$$\sum_{i \in \mathcal{P}} u_i = \prod_{i \in \mathcal{P}} k_i \qquad \text{and} \qquad \sum_{i \in \mathcal{P}} \tilde{v}_i = \prod_{i \in \mathcal{P}} \frac{\phi_i}{k_i}.$$

Consistency Check:

- Each party P_i computes $R_i := u_i \cdot G$ and commits R_i
- Upon being notified of all other parties' commitments, each party P_j open commitment of R_j. Then each party computes

$$R := \sum_{j \in \mathcal{P}} R_j.$$

- Each party calculates $\Gamma_i^1 := \tilde{v}_i \cdot R$ and commit Γ_i^1.
- Upon being notified of all other parties' commitments, each party P_j open commitment of (ϕ_i, Γ_i^1).

- Each party P_i sets $\{u_i, \tilde{v}_i\} = \psi_i^{\lceil \log_2(t) \rceil + 1}$ and so

$$\sum_{i \in \mathcal{P}} u_i = \prod_{i \in \mathcal{P}} k_i \quad \text{and} \quad \sum_{i \in \mathcal{P}} \tilde{v}_i = \prod_{i \in \mathcal{P}} \frac{\phi_i}{k_i}.$$

Consistency Check:

- Each party P_i computes $R_i := u_i \cdot G$ and commits R_i
- Upon being notified of all other parties' commitments, each party P_j open commitment of R_j. Then each party computes

$$R := \sum_{j \in \mathcal{P}} R_j.$$

- Each party calculates $\Gamma_i^1 := \tilde{v}_i \cdot R$ and commit Γ_i^1.
- Upon being notified of all other parties' commitments, each party P_j open commitment of (ϕ_i, Γ_i^1).
- Each party calculates

$$\phi := \prod_{j \in \mathcal{P}} \phi_j,$$

and aborts if the pad is equal to zero, or if

$$\sum_{j \in \mathcal{P}} \Gamma_j^1 \neq \phi \cdot G.$$

- Each party P_i sets $\{u_i, \tilde{v}_i\} = \psi_i^{\lceil \log_2(t) \rceil + 1}$ and so

$$\sum_{i \in \mathcal{P}} u_i = \prod_{i \in \mathcal{P}} k_i \quad \text{and} \quad \sum_{i \in \mathcal{P}} \tilde{v}_i = \prod_{i \in \mathcal{P}} \frac{\phi_i}{k_i}.$$

Consistency Check:

- Each party P_i computes $R_i := u_i \cdot G$ and commits R_i
- Upon being notified of all other parties' commitments, each party P_j open commitment of R_j. Then each party computes

$$R := \sum_{j \in \mathcal{P}} R_j.$$

- Each party calculates $\Gamma_i^1 := \tilde{v}_i \cdot R$ and commit Γ_i^1.
- Upon being notified of all other parties' commitments, each party P_j open commitment of (ϕ_i, Γ_i^1).
- Each party calculates

$$\phi := \prod_{j \in \mathcal{P}} \phi_j,$$

and aborts if the pad is equal to zero, or if

$$\sum_{j \in \mathcal{P}} \Gamma_j^1 \neq \phi \cdot G.$$

- Each party P_i computes $v_i := \tilde{v}_i / \phi$ and takes (u_i, v_i, R) as its output.

Setup

$$
\begin{array}{cc}
\underline{P_i} & \underline{P_j} \\
p_i(x) \leftarrow \text{poly-degre}(t-1) & \\
p_i(j) & \xrightarrow{\quad p_i(j) \quad} \\
p(i) = \sum_{j \in P} p_j(i) & \xleftarrow{\quad p_j(i) \quad} \\
T_i := p(i) \cdot G & \\
c_i := \text{commit}(T_i) & \xrightarrow{\quad c_i \quad} \\
 & \xleftarrow{\quad c_j \quad} \\
 & \xrightarrow{\quad T_i \quad} \\
\text{open}(c_j) \stackrel{?}{=} T_j & \xleftarrow{\quad T_j \quad} \\
\pi_i := \text{DLZKP}(p(i)) & \xrightarrow{\quad \pi_i \quad} \\
\text{ZKVerify}(\pi_j) \stackrel{?}{=} 1 & \xleftarrow{\quad \pi_j \quad}
\end{array}
$$

$$\forall x \in [1, n-t-1]$$

$$J^x = [x, x+t] \text{ and } J^{x+1} = [x+1, x+t+1]$$

$$\sum_{j \in J^x} \lambda_j^{J^x}(0) \cdot T_j \stackrel{?}{=} \sum_{j \in J^{x+1}} \lambda_j^{J^{x+1}}(0) \cdot T_j$$

Signing

$$
\begin{array}{ccc}
\underline{P_i} & & \underline{P_j} \\
\text{Call } (u_i, v_i, R) \leftarrow \mathcal{F}_{\text{inv}}^{n,t} & & \\
sk_i := \lambda_i^P(0) \cdot p(i) & & \\
\{sk_i, v_i\} \longrightarrow & \boxed{\mathcal{F}_{\text{2PMul}}^2} & \longleftarrow \{v_j, sk_j\} \\
\{w_i^{j,1}, w_i^{j,2}\} \longleftarrow & & \longrightarrow \{w_j^{i,1}, w_j^{i,2}\}
\end{array}
$$

$$w_i^{j,1} + w_j^{i,1} = sk_i \cdot v_j$$
$$w_i^{j,2} + w_j^{i,2} = sk_j \cdot v_i$$

$$w_i := sk_i \cdot v_i + \sum_{j \in P/\{i\}} w_i^{j,1} + w_i^{j,2}$$

$$\Gamma_i^2 := v_i \cdot pk - w_i \cdot G$$
$$\Gamma_i^3 := w_i \cdot R$$
$$c_i := \text{commit}(\Gamma_i^2, \Gamma_i^3) \qquad \xrightarrow{c_i}$$
$$\xleftarrow{c_j}$$
$$\xrightarrow{\Gamma_i^2, \Gamma_i^3}$$
$$\text{open}(c_j) \stackrel{?}{=} (\Gamma_j^2, \Gamma_j^3) \qquad \xleftarrow{\Gamma_j^2, \Gamma_j^3}$$
$$\sum_{j \in P} \Gamma_j^2 \stackrel{?}{=} 0$$
$$\sum_{j \in P} \Gamma_j^3 \stackrel{?}{=} \text{pk}$$

$$\text{sig}_i := H(m) \cdot v_i + w_i \cdot r_x$$

$$\text{sig} = \sum_{i \in P} \text{sig}_i$$

Performance of DKLs-19

- Setup: 5 rounds (key generation) + 3 rounds ("auxiliary setup")
- Signing: $6 + \lceil \log t \rceil$ rounds (only the last is online)

Performance of DKLs-19

- Setup: 5 rounds (key generation) + 3 rounds ("auxiliary setup")
- Signing: $6 + \lceil \log t \rceil$ rounds (only the last is online)

	Signing		Setup	
Protocol	$t = 2$	$t = 20$	$n = 2$	$n = 20$
This Work	8.6	31.5	48.5	242
[GG18]	77	509	–	–
[LNR18]	304	5194	∼11000	∼28000
[BGG17]	∼650	∼1500	–	–
[GGN16]	205	1136	–	–
[Lin17]	36.8	–	2435	–
[DKLs18]	3.8	–	43.4	177

Table 4: **Wall-clock Time Comparison to Other Works**. Note that benchmarking environments are non-identical, but all benchmarks are networkless or over LAN.