New Constructions of Traceable Range Proofs: Towards Multiple Regulation and Joint Regulation

Wulu Li*, Lei Chen*, Xin Lai*, and Xiao Zhang*
*Onething Technologies Co., Ltd.,
Shenzhen, China,
lwlmax@126.com

***Abstract*—Traceable range proof (TRP) plays a major role in the construction of regulatable privacy-preserving blockchains, as it empowers regulators with traceability of the hidden amount in each transaction. In this paper, we give new constructions of TRPs with improved efficiency and more regulatory functions. In particular, we introduce sTBoRP: a simplified traceable Borromean range proof directly from Borromean ring signature without additional validity proofs for tracing keys, sTBoRP can be applied for multiple regulation between different regulators, and can be further modified to be secure against malicious regulators. Moreover, we introduce jTBuRP: a modified traceable Bulletproofs range proof to support joint regulation against collusion attack of malicious regulators, by improving the generation algorithm of tracing keys. We also give the security proofs for both schemes and give the comparisons of efficiency and security.**

***Index Terms*—Traceable range proof, regulatable blockchain, multiple regulation, joint regulation, privacy-preserving**

I. Introduction

Privacy-preserving techniques in blockchain theory have been developed in this decade to provide a potential replacement of traditional blockchain-based cryptocurrencies such as Bitcoin [1] and Ethereum [2]. Privacy-preserving cryptocurrencies, represented by Monero [3] and Zerocash [4], have realized anonymous and confidential transactions, which can protect the transaction amount and users' identities, making them suitable in privacy applications such as salary, donation, bidding, taxation, etc. A series of works have been proposed during these years such as Confidential Transaction [5], Mimblewimble [6], Dash [7], Monero [3] and Zerocash [4], etc. Among them, Monero uses techniques from Cryptonote [3], Ring-CT [8], Bulletproofs [9] as building blocks, it uses linkable ring signature scheme to hide the identity of initiator, uses Diffie-Hellman key exchange scheme to hide the identity of recipient and uses range proof (Borromean [8], Bulletproofs [9]) to hide the the amount of transaction.

However, the privacy-preserving cryptocurrencies are not traceable, which may cause abuse of privacy and facilitate illegal transactions by malicious users. It is crucial to develop new regulatory mechanism to realize traceability of users' identities and transaction amount. To solve this issue, a recent work by Li *et al.* [10] proposes the first fully regulatable privacy-preserving blockchain against malicious regulators, their construction contains traceable and linkable ring signature scheme (TLRS), traceable range proofs (TBoRP, TBuRP) and traceable scheme of long-term addresses. Their work is an effective approach to overcome the regulatory barriers on privacy-preserving cryptocurrency. In the construction of TBoRP, an extra validity proof of tracing keys is needed to prevent traceability attack (escaping from regulation), which requires extra storage and more time for generation and verification, making TBoRP less efficient than the original Borromean range proof. So it is necessary to construct new TBoPR with more compact size and less time for generation and verification, to support the future application of cryptocurrency for high TPS (transactions per second). Moreover, both TBoRP and TBuRP have only one regulator, cannot support regulatory functions such as multiple regulation and joint regulation, which are needed in transnational cryptocurrency regulations between banks and different national regulators. In this paper, we give new constructions of TRPs to realize multiple regulation and joint regulation.

A. Regulatory Models

1) Multiple Regulation: Multiple regulation means that there are m independent regulators (mutually distrustful) in the blockchain, who generate the trapdoors respectively to trace the transaction amount, without extra communication or computation between them. A trivial solution to fulfill this task is to run the traceable range proof for m times (with each regulator's trapdoor parameter respectively), which is rather inefficient for storage and time. It remains an open problem to realize multiple regulation between mutually distrustful regulators within one traceable range proof. We give a simplified construction of TBoRP in this paper to solve this problem.

2) Joint Regulation: Joint regulation means that there are m different independent branch regulators and a chief regulator (can be considered as the central bank and branch banks), every branch regulator generates a trapdoor to trace the partial amount in each transaction, while the chief regulator possesses no trapdoors and can only receive the tracing results from all branch regulators to recover the total amount. Moreover, we also introduce joint regulation against collusion attack of malicious (branch) regulators, which means any number ($< m$) of colluded branch regulators cannot recover any bit of the hidden amount, unless all branch regulators are corrupted. In this paper, we introduce a modification of TBuRP to realize the regulatory function of joint regulation with security against collusion attack.

B. Our Contributions

In this paper, we propose sTBoRP (simplified traceable Borromean range proof) and jTBuRP (traceable Bulletproofs range proof for joint regulation), to realize the functionalities of multiple regulation and joint regulation respectively, we give a brief introduction of them:

1) sTBoRP: In the construction of sTBoRP, we use Pedersen commitment $c = g^x h^a$ to hide the transaction amount a, every regulator $\mathcal{R}_j$ generates his trapdoor y_j and computes the corresponding trapdoor parameter $h_j = g^{y_j}, j = 1, \cdots, m$. For a's binary expansion $a = a_0 + 2a_1 + \cdots + 2^{n-1}a_{n-1}$, for every bit $i = 0, \cdots, n-1$, prover computes m different tracing keys $TK_{i,j}, j = 1, \cdots, m$, prover also generates a ring L^i_{PK} with two elements and generates the Borromean ring signature for n rings. We improve the TBoRP by removing the additional validity proofs of all $TK_{i,j}$, which is used in TBoRP to prevent traceability attack. The verifier only need to check the validity of Borromean ring signature and the correctness of binary expansion. All the regulators can trace $a_i = 0 \ or \ 1$ for every $i = 0, \cdots, n-1$ to compute the total amount $a = a_0 + 2a_1 + \cdots + 2^{n-1}a_{n-1}$ by usage of trapdoor y_j and $TK_{i,j}$, without extra communication or computation.

We can prove that any PPT adversary (including malicious regulators) cannot break the completeness, soundness of sTBoRP. In addition, sTBoRP can be further modified by adding n key-images to achieve traceability against malicious regulators. Compared to TBoRP, the proof size of sTBoRP ($m = 1$) is reduced from ($2n$, $2n$+5) to ($2n$, $2n$+2), where the $(\cdot, \cdot)$ refers to number of elements in $(\mathbb{G}, \mathbb{Z}_q^*)$, the generation computation of sTBoRP is reduced from ($7n$+2, $4n$-1) to ($6n$+1, $3n$+2), the verification computation of sTBoRP is reduced from ($6n$+5, $6n$+1) to ($5n$+2, $6n$+1), where $(\cdot, \cdot)$ refers to number of exponentiations and multiplications in $\mathbb{G}$. Moreover, soundness of sTBoRP (against malicious regulator) is better than TBoRP (against honest regulator).

2) jTBuRP: In the construction of jTBuRP, similar to TBuRP, for different generators $\mathbf{g} = (g_0, \cdots, g_{n-1})$ generated independently by system, all branch regulators reach a consensus about the bit partition of transaction amount, and generate their trapdoors, then compute the corresponding trapdoor parameters. For a's commitment $c = h^x g^a$ and binary expansion $a = a_0 + 2a_1 + \cdots + 2^{n-1}a_{n-1}$, the prover computes tracing keys with their validity proofs, then finishes the rest of Bulletproofs, note that the major improvement of jTBuRP is the merge operations in the tracing key generation algorithm, which brings jTBuRP with security against collusion attack. The verifier checks the validity of Bulletproofs and the validity proofs of all tracing keys. The branch regulators send their tracing results to the chief regulator, who can recover the hidden amount a, while any number of branch regulators ($< m$) cannot recover any bit of a. An optional choice of jTBuRP is to remove the chief regulator and use MPC to recover a between all branch regulators, it is technically feasible but is not the focus of this paper as we only focus on the construction and modification of traceable range proofs.

In jTBuRP, soundness holds for any PPT adversary (including malicious regulators), traceability only holds for PPT adversary without possession of the trapdoors. And the number of trapdoors and tracing keys can be adjusted for different regulatory requirements in application, which gives a potential replacement of Bulletproofs-based cryptocurrency for joint regulation.

C. Related Works

1) Range Proofs: Range proof is a special zero-knowledge proof to prove a committed hidden amount a lies within a certain range $[0, 2^n - 1]$ without revealing the amount. The Pedersen-commitment-based range proofs are used in Monero system. In 2016, Neother *et al.* [8] gave the Borromean range proof, building from the Borromean ring signature [11], with linear proof size to the binary length of range. In 2018, Bünz *et al.* [9] introduced Bulletproofs, an efficient non-interactive zero-knowledge proof protocol with short proofs and without a trusted setup, the proof size is only logarithmic to the witness size and it is used in projects such as Monero, DERO [12]. There are also privacy-preserving blockchain systems such as Qiusqius [13], Zether [14] with different commitments and range proofs. Moreover, range proof can also be built from zero-knowledge for arithmetic circuits, including [15]–[20].

2) Traceable Range Proofs: Traceable range proof (TRP) is a special variant of range proof, there is a regulator who can use trapdoors to trace the hidden amount. The zero-knowledge property of traceable range proof only holds for users without possession of trapdoors. Li *et al.* [10] proposes the first traceable range proof, their work contains traceable Borromean range proofs (TBoRP) and traceable Bulletproofs range proofs (TBuRP), which are the potential replacements in Monero-type cryptocurrency to realize the regulatory function. To the best of our knowledge, there are no traceable range proofs with regulatory functions such as multiple regulation and joint regulation.

D. Organization

In section II we give some preliminaries; in section III we give the construction and security proofs of sTBoRP; in section IV we give the construction and security proofs of jTBuRP; in section V we give the analysis and comparison of the schemes; in section VI we give the conclusion.

II. Preliminaries

In this paper, we use multiplicative cyclic group $\mathbb{G}$ to represent elliptic group with prime order $|\mathbb{G}| = q$, g is the generator of $\mathbb{G}$, group multiplication is $g_1 \cdot g_2$ and exponentiation is g^a. We use $H(\cdot)$ to represent hash function and $negl$ to represent negligible functions. For verifiers, 1 is for *accept* and 0 is for *reject*. For adversaries, PPT means probabilistic polynomial time. The extended DDH assumption means any PPT adversary cannot distinguish $(g^a, h_1^a, \cdots, h_m^a)$ from $(g^a, r_1, \cdots, r_m)$, where r_j is uniformly sampled from $\mathbb{G}$. The hardness of discrete logarithm problem means that any PPT adversary cannot compute x from g^x. Oracle $\mathcal{RO}$ refers to the random oracle.

A. Zero-knowledge Proofs

Zero-knowledge proof system is a proof system (P, V) in which a prover proves to the verifier that he has a certain knowledge but does not reveal the knowledge itself. The formal definition is that given language L and relation R, for $\forall x \in L$, there exists a witness w such that $(x, w) \in R$, to prove $x \in L$ without disclosing w. The transcript between prover and verifier is $\langle P(x, w), V(x)\rangle$, the proof is correct (or wrong) if $\langle P(x, w), V(x)\rangle = 1(or\ 0)$. The security notions of zero-proof system contains *completeness*, *soundness*, *zero-knowledge*:

Definition 1 (Completeness): (P, V) has **completeness** for any non-uniform polynomial time adversary $\mathcal{A}$,

$$\begin{aligned}\mathsf{Pr}[(x, w) \leftarrow \mathcal{A}(1^\lambda) : (x, w) \notin R\ or\ \langle P(x, w), V(x)\rangle = 1]\\ = 1 - negl.\end{aligned}$$

When the probability equals 1, then (P, V) has perfect completeness.

Definition 2 (Soundness): (P, V) has **soundness** for any non-uniform polynomial time adversary $\mathcal{A}$ and $x \notin L$,

$$\mathsf{Pr}[(x, s) \leftarrow \mathcal{A}(1^\lambda) : \langle P(x, w), V(x)\rangle = 1] = negl.$$

In Σ protocols with Fiat-Shamir transformation in the random oracle model, we use the notion of **special soundness**, that is, for a 3-round interactive proof protocol, if a non-uniform polynomial time adversary $\mathcal{A}$ can generate 2 valid proofs $(x, c, e_1, s_1), (x, c, e_2, s_2)$, then there exists a extraction algorithm Ext which can extract a witness $(x, w) \in R$, where c represents the commitment, e_is are challenges and s_is are responses.

Definition 3 (Zero-knowledge): (P, V) has **perfect** (or **computational**) **zero-knowledge**, for any non-uniform polynomial time (or PPT) adversary $\mathcal{A}$,

$$\begin{aligned}&\mathsf{Pr}[(x, w) \leftarrow \mathcal{A}(1^\lambda); tr \leftarrow \langle P(x, w), V(x, \rho)\rangle :\\ &(x, w) \in R\ and\ \mathcal{A}(tr) = 1]\\ &= (or \approx_c)\mathsf{Pr}[(x, w) \leftarrow \mathcal{A}(1^\lambda); tr \leftarrow S(x, \rho) :\\ &(x, w) \in R\ and\ \mathcal{A}(tr) = 1].\end{aligned}$$

In Fiat-Shamir-based protocol, the randomness of ρ is from the output of hash function, it is said to be **public coin** and the protocol is **honest-verifier zero-knowledge**.

1) Pedersen Commitment: Pedersen commitment [21] was proposed in 1991, for elliptic curve $(\mathbb{G}, q = |\mathbb{G}|, g, h)$, where g is a generator of $\mathbb{G}$, h is a random element with discrete logarithm unknown to anyone.

Definition 4 (Pedersen commitment): The Pedersen commitment for a is $c = g^x h^a$, where $x \in \mathbb{Z}_q^*$ is a blinding element. Under the hardness of discrete logarithm, Pedersen commitment has the following properties:

– (Hiding) Any (computational unbounded) adversary $\mathcal{A}$ cannot distinguish $c = g^x h^a$ from $c' = g^{x'} h^{a'}$.
– (Binding) Any PPT adversary $\mathcal{A}$ cannot generate another secret a' binding with $c = g^x h^a = g^{x'} h^{a'}$.
– (Homomorphic) Given $c_1 = g^x h^a, c_2 = g^y h^b$, then $c_1 \cdot c_2 = g^{x+y} h^{a+b}$ is a new commitment for $a + b$.

2) Proof of Committed Values: For commitment $c = \prod_{i=1}^n g_i^{x_i}$, we can prove the knowledge of $x_1, \cdots, x_n$ without revealing them by proof of committed values:

1. Prover generates $r_1, \cdots, r_n \in \mathbb{Z}_q^*$ uniformly, computes $e = H(\prod_{i=1}^n g_i^{r_i})$.
2. Prover computes $z_i = r_i + ex_i$ for $i = 1, \cdots, n$, output proof $\pi(c) = (z_1, \cdots, z_n, e)$.
3. Verifier checks $e \stackrel{?}{=} H(\prod_{i=1}^n g_i^{z_i}/c^e)$.

Proof of committed values is an extension of Schnorr signature ($n = 1$) with perfect completeness, special soundness and honest verifier zero-knowledge.

B. Range proofs

1) Borromean Range Proof: Borromean range proof [8] is used in Monero to provide the validity proof of transaction amount ($a \in [0, 2^n - 1]$) by usage of Borromean ring signature [11] and Pedersen commitment, the detailed description of Borromean range proof is in [8].

2) Bulletproofs Range Proof: Bulletproofs, proposed by Bünz *et al.* [9] in 2018, is an efficient zero-knowledge with $O(\log n)$ proof size, and is widely used in inner-product argument, range proof and proof for arithmetic circuits. The Bulletproofs range proof also uses Pedersen commitment, the description of Bulletproofs is in [9].

C. Traceable Range proofs

1) Traceable Borromean range proof: Traceable Borromean range proof [10] provides the validity proof and traceability of transaction amount ($a \in [0, 2^n - 1]$) by usage of Borromean ring signature and Pedersen commitment, please refer to [10] for detailed description of TBoRP.

2) Traceable Bulletproofs range proof: Traceable Bulletproofs range proof [10] provides the same functionality as TBoRP, with different construction and more flexible parameter selection, please refer to [10] for detailed description of TBuRP.

D. Security Models

The definitions of completeness, soundness and zero-knowledge are in II.A. Considering the existence of regulator, who can trace the amounts of transactions, zero-knowledge only holds for someone not possesses the trapdoor, while the completeness and soundness remains the same as in range proof, for any PPT adversary $\mathcal{A}$. For traceable range proof, we need another security concept called **traceability**. Since traceable range proof enables regulator with ability to trace the hidden amounts of transactions, for any PPT adversary $\mathcal{A}$ (without possession of trapdoors), it is necessary that he cannot escape from regulation (making his transaction amount untraceable). We give the formal definition of traceability as follows:

Definition 5 (Traceability): Traceability for traceable range proof is defined in the following game between the simulator $\mathcal{S}$ and the adversary $\mathcal{A}$, simulator $\mathcal{S}$ runs Setup to provide public parameters for $\mathcal{A}$, $\mathcal{A}$ is given access to oracle $\mathcal{RO}$. $\mathcal{A}$ generates a commitment c for a hidden value a and the range proof $\pi(c)$, $\mathcal{A}$ wins the game if:

1. $\mathsf{Verify}(c, \pi(c)) = 1$.
2. $\mathsf{Trace}(\pi(c), trapdoors) \neq a$.

We give the advantage of $\mathcal{A}$ in traceability attack as follows:

$$\mathrm{Adv}_{\mathcal{A}}^{trace} = \Pr[\mathcal{A} \text{ wins}].$$

A traceable range proof is traceable if for any PPT adversary $\mathcal{A}$, $\mathrm{Adv}_{\mathcal{A}}^{trace} = negl$.

Note that sTBoRP can be further modified to achieve traceability against malicious regulators, by adding key-images to the proof, we will introduce the modification in III.C.

III. Simplified TBoRP for Multiple Regulation

In this section we introduce the construction and security proof of simplified traceable Borromean range proof (sTBoRP), and introduce its application for multiple regulation. Moreover, we give the modification of sTBoRP to achieve traceability against malicious regulators.

A. Construction

In the construction of sTBoRP, similar to TBoRP, we use Pedersen commitment and bit expansion of amount, then add tracing keys bitwise into the proof, and remove the validity proofs for tracing keys in TBoRP, the sets of public keys for Borromean ring signature is also modified to support multiple regulation. Every regulator can use his trapdoor and tracing keys to recover the hidden amount without additional communication or computation.

We give the introduction of sTBoRP in the following:

$\mathsf{Par} \leftarrow \mathsf{Setup}(\lambda)$:

1. System chooses elliptic curve $\mathbb{G}$ with prime order q and samples independent generators $g, h \leftarrow \mathbb{G}$. For $j = 1, \cdots, m$, regulator $\mathcal{R}_j$ generates $y_j \in \mathbb{Z}_q^*$ as his trapdoor, computes $h_j = g^{y_j}$, system outputs $(\mathbb{G}, q, g, h, h_1, \cdots, h_m)$ as the public parameters.

$(c, \beta, SK, \{(c_i, c_i'), TK_{i,j}\}_{j\in[1,m]}^{i\in[0,n-1]}) \leftarrow \mathsf{Gen}(\mathsf{Par}, a)$:

1. According to the public parameters and amount $a \in [0, 2^n - 1]$, prover Alice samples $x \in \mathbb{Z}_q^*$ uniformly, computes $c = g^x h^a$ as the commitment;
2. Alice computes the binary expansion $a = a_0 + \cdots + 2^{n-1}a_{n-1}, a_i = 0, 1$ for $i = 0, \cdots, n-1$, samples $x_0, \cdots, x_{n-1}$ uniformly, computes $\beta = x - x_0 - \cdots - x_{n-1}$;
3. For every $i = 0, \cdots, n-1$, Alice computes $c_i = g^{x_i} h^{2^i a_i}, c_i' = g^{x_i} h^{2^i a_i - 2^i}$, outputs $\{c_i, c_i'\}_{i\in[0,n-1]}$;
4. For every $i = 0, \cdots, n-1$, $j = 1, \cdots, m$, Alice computes $TK_{i,j} = h_j^{x_i}$, then computes $e_k = H(c_0, \cdots, c_{n-1}, \{TK_{i,j}\}_{j\in[1,m]}^{i\in[0,n-1]}, k)$ for $k = 1, \cdots, m$;
5. Alice computes $L_{PK}^i = \{c_i \cdot \prod_{j=1}^m TK_{i,j}^{e_j}, c_i' \cdot \prod_{j=1}^m TK_{i,j}^{e_j}\}$ for $i = 0, \cdots, n-1$;
6. Alice outputs $(c, \beta, \{(c_i, c_i'), TK_{i,j}\}_{j\in[1,m]}^{i\in[0,n-1]})$ and retains $(a, SK = (x_0, \cdots, x_{n-1}), \{L_{PK}^i\}_{i\in[0,n-1]})$.

$\pi_{sTBo}(c) \leftarrow \mathsf{Prove}(SK, c, L_{PK})$:

1. For message μ, Alice runs the Borromean ring signature with $L_{PK} = \{L_{PK}^0, \cdots, L_{PK}^{n-1}\}$, computes $\sigma \leftarrow \mathsf{Rsign}(SK, \mu, L_{PK})$;
2. Alice outputs the sTBoRP proof $\pi_{sTBo}(c) = (c, \beta, \{c_i, c_i'\}_{i\in[0,n-1]}, \{TK_{i,j}\}_{j\in[1,m]}^{i\in[0,n-1]}, \sigma)$.

$1/0 \leftarrow \mathsf{Verify}(\pi_{sTBo}(c))$:

1. Verifier computes $e_k = H(c_0, \cdots, c_{n-1}, \{TK_{i,j}\}_{j\in[1,m]}^{i\in[0,n-1]}, k)$ for $k = 1, \cdots, m$;
2. Verifier computes $L_{PK}^i = \{c_i \cdot \prod_{j=1}^m TK_{i,j}^{e_j}, c_i' \cdot \prod_{j=1}^m TK_{i,j}^{e_j}\}$ for $i = 0, \cdots, n-1$;
3. For every $i = 0, \cdots, n-1$, verifier checks $c_i/c_i' \stackrel{?}{=} h^{2^i}$, then checks $g^\beta \cdot \prod_{i=0}^{n-1} c_i \stackrel{?}{=} c$;
4. Verifier checks the validity of Borromean ring signature σ, if all passed then outputs 1, otherwise outputs 0.

$a^* \leftarrow \mathsf{Trace}(y_j, \pi_{sTBo}(c))$, $j = 1, \cdots, m$:

1. For every $i = 0, \cdots, n-1$, $\mathcal{R}_j$ computes $c_i^{y_j}$;
2. For every $i = 0, \cdots, n-1$, if $c_i^{y_j} = TK_{i,j}$ then outputs $a_i^* = 0$, otherwise outputs $a_i^* = 1$;
3. $\mathcal{R}_j$ outputs $a^* = a_0^* + \cdots + 2^{n-1}a_{n-1}^*$ as the result.

Alg. 1. sTBoRP

Note that the Borromean ring signature Rsign (for n rings) uses $g \cdot \prod_{j=1}^m h_j^{e_j}$ as the generator (basis element), correctness and security of sTBoRP will be proved in the next subsection. Meanwhile, the message μ contains $c, \{c_i, c_i'\}_{i\in[0,n-1]}$ and other information related with the transaction, similar to Monero. Moreover, all regulators can recover the hidden amount by their trapdoors respectively. Actually, c_i' can be removed from the proof since $c_i' = c_i/h^{2^i}$.

sTBoRP also supports joint regulation, by generating $\{TK_i\}_{i\in[0,n-1]}$, which is a subset of $\{TK_{i,j}\}_{j\in[1,m]}^{i\in[0,n-1]}$, and modifying the generator in each ring L_{PK}^i, but sTBoRP cannot resist collusion attack, as every branch regulator can recover some bits of the hidden amount, the detailed description of sTBoRP for joint regulation is in the full version.

B. Correctness and Security

1) Proof of Correctness:

Theorem 6 (Correctness of sTBoRP): For an honest user Alice, she can complete sTBoRP successfully and all regulators can trace the hidden amount correctly and independently.

Proof: According to the binary expansion $a = a_0 + \cdots + 2^{n-1}a_{n-1}$ of a and $c_i = g^{x_i}h^{2^i a_i}, c_i' = g^{x_i}h^{2^i a_i - 2^i}$, we know there is only one element in $L_{PK}^i = \{c_i \cdot \prod_{j=1}^m TK_{i,j}^{e_j}, c_i' \cdot \prod_{j=1}^m TK_{i,j}^{e_j}\}$, which is a power of $g \cdot \prod_{j=1}^m h_j^{e_j}$ known by Alice, then Alice can use the secret keys $SK = (x_0, \cdots, x_{n-1})$ ($SK_i = x_i$ with corresponding $PK_i = (g \cdot \prod_{j=1}^m h_j^{e_j})^{x_i}$) to finish the Borromean ring signature for $L_{PK} = \{L_{PK}^0, \cdots, L_{PK}^{n-1}\}$. Besides, we know that $g^\beta \cdot \prod c_i = c$ and $c_i/c_i' = h^{2^i}$ from the Gen algorithm. When $a_i = 0$, we know $c_i = g^{x_i}h^{2^i a_i} = g^{x_i}$, $TK_{i,j} = h_j^{x_i} = c_i^{y_j}$ for $j = 1, \cdots, m$. When $a_i = 1$, we know $TK_{i,j} = h_j^{x_i}, c_i^{y_j} = (g^{x_i}h^{2^i})^{y_j} = h_j^{x_i + 2^i y_j}$, then $c_i^{y_j} = TK_{i,j}$ iff $y_j = 0$, which happens with negligible probability, then all the regulators can recover a correctly and independently, so we get the correctness of sTBoRP. □

2) Proof of Soundness: Completeness is easily obtained from the correctness of sTBoRP, here we prove the soundness of sTBoRP, which means any PPT adversary with possession of all trapdoors cannot generate a valid $\pi_{sTBo}(c)$ for $c = g^x h^a$ and $a \notin [0, 2^n - 1]$, under the hardness of discrete logarithm and the unforgeability of Borromean ring signature.

Theorem 7 (Soundness of sTBoRP): sTBoRP has computational soundness for any PPT adversary $\mathcal{A}$, including malicious regulator.

Proof: For $c = g^x h^a$ with $a \notin [0, 2^n - 1]$, assume $\mathcal{A}$ (with possession of all trapdoors) successfully outputs a valid proof $\pi_{sTBo}(c)$, then we have $g^\beta \cdot \prod_{i=0}^{n-1} c_i = c$ and $c_i/c_i' = h^{2^i}$. Without loss of generality, we set $c_i = g^{x_i} h^{b_i}$, then $c_i' = g^{x_i} h^{b_i - 2^i}$, since $a \notin [0, 2^n - 1]$, we know there exists at least one $l \in \{0, \cdots, n-1\}$, satisfying $b_l \neq 0$ and $b_l \neq 2^l$, otherwise $g^\beta \cdot \prod_{i=0}^{n-1} c_i = g^{\sum x_i} h^{\sum b_i} = g^x h^a$ with $\sum b_i \in [0, 2^n - 1]$, then $h^{a - \sum b_i} = g^{\beta + \sum x_i - x}$, so $\mathcal{A}$ gets a non-trivial relation between g and h, which happens with negligible probability. Then we have:

$$L_{PK}^l = \{c_l \cdot \prod_{j=1}^m TK_{l,j}^{e_j}, c_l' \cdot \prod_{j=1}^m TK_{l,j}^{e_j}\},$$

where $e_k = H(c_0, \cdots, c_{n-1}, \{TK_{i,j}\}_{j \in [1,m]}^{i \in [0,n-1]}, k)$ for $k = 1, \cdots, m$, since $\mathcal{A}$ knows all the relations between g and h_j for $j = 1, \cdots, m$, we can set $TK_{l,j} = g^{s_j} h^{t_j}$, then $L_{PK}^l = \{g^{x_l + \sum s_j e_j} h^{b_l + \sum t_j e_j}, g^{x_l + \sum s_j e_j} h^{b_l - 2^l + \sum t_j e_j}\} = \{PK_l, PK_l'\}$ with $b_l \neq 0$ and $b_l \neq 2^l$. Since the generator for Borromean ring signature is $g \cdot \prod_{j=1}^m h_j^{e_j} = g^{1 + \sum y_j e_j}$, from the unforgeability of Borromean ring signature, we know there is at least one public key from $\{PK_l, PK_l'\}$ (without loss of generality we set PK_l) satisfying $g^{x_l + \sum s_j e_j} h^{b_l + \sum t_j e_j} = (g^{1 + \sum y_j e_j})^z$, with z known to $\mathcal{A}$. If $b_l + \sum t_j e_j \neq 0$ then $\mathcal{A}$ gets a non-trivial relation between g and h, which happens with negligible probability. Then we have $b_l + \sum t_j e_j = 0$, if there exists $k \in [1, m]$ *s.t.* $t_k \neq 0$, then $e_k = (-b_l - \sum_{j \neq k} t_j e_j) t_k^{-1}$, which means e_k can be pre-computed before $\mathcal{A}$ run the hash function, which also happens with negligible probability, then we have $t_j = 0$ for $j = 1, \cdots, m$, and then $b_l = 0$, which contradicts with the assumptions before, then we get the soundness of sTBoRP against malicious regulators. □

3) Proof of Zero-knowledge:

Theorem 8 (Zero-knowledge of sTBoRP): sTBoRP is computational zero-knowledge for any PPT adversary $\mathcal{A}$ (without possession of trapdoors).

Proof: For every $i = 0, \cdots, n-1$, we consider the effect that $\{TK_{i,j}\}_{j \in [1,m]}$ being added into the proof, and prove that $(c_i, \{TK_{i,j}\}_{j \in [1,m]})$ is computational indistinguishable from uniform distribution when $a_i = 0$ or 1. Formally, we prove for $c_i = g^{x_i} h^{2^i a_i}, c_i' = g^{x_i} h^{2^i a_i - 2^i}$ with $c_i/c_i' = h^{2^i}$ being a constant, any PPT adversary $\mathcal{A}$ cannot distinguish uniform distribution $U = (r, r_1, \cdots, r_m)$ from $(c_i, TK_{i,1}, \cdots, TK_{i,m}) = (g^{x_i}, h_1^{x_i}, \cdots, h_m^{x_i})$ (when $a_i = 0$) or $(c_i, TK_{i,1}, \cdots, TK_{i,m}) = (g^{x_i} h^{2^i}, h_1^{x_i}, \cdots, h_m^{x_i})$ (when $a_i = 1$), where U is sampled uniformly from $\mathbb{G}^{m+1}$.

Actually, we know that $(g^{x_i}, h_1^{x_i}, \cdots, h_m^{x_i})$ and $(g^{x_i}, r_1, \cdots, r_m)$ are computational indistinguishable for uniformly generated $x_i \in \mathbb{Z}_q^*$, under the extended DH assumption. For g being a generator of $\mathbb{G}$, the distribution of $(g^{x_i}, r_1, \cdots, r_m)$ and $(r, r_1, \cdots, r_m)$ are identical. Let constant $u = h^{2^i}$, we know that the distribution of $(r, r_1, \cdots, r_m)$ and $(ru, r_1, \cdots, r_m)$ are identical. Again from the extended DH assumption, we know $(ru, r_1, \cdots, r_m)$ and $(g^{x_i} u, h_1^{x_i}, \cdots, h_m^{x_i})$ are computational indistinguishable. Then we have the following relations:

$$(g^{x_i}, h_1^{x_i}, \cdots, h_m^{x_i}) \approx_c (r, r_1, \cdots, r_m) = U$$
$$= (ru, r_1, \cdots, r_m) \approx_c (g^{x_i} u, h_1^{x_i}, \cdots, h_m^{x_i}).$$

Where $g, h, h_1, \cdots, h_m, u \in \mathbb{G}$ are constants, $U \leftarrow \mathbb{G}^{m+1}, x_i \leftarrow \mathbb{Z}_q^*$ are sampled uniformly at random.

Since $(g^{x_i}, h_1^{x_i}, \cdots, h_m^{x_i}) = (c_i, TK_{i,1}, \cdots, TK_{i,m})_{a_i=0}$ and $(g^{x_i} u, h_1^{x_i}, \cdots, h_m^{x_i}) = (c_i, TK_{i,1}, \cdots, TK_{i,m})_{a_i=1}$, we know they are all computational indistinguishable from $U = (r, r_1, \cdots, r_m)$ for any PPT adversary $\mathcal{A}$ without possession of trapdoors. Since all x_is are uniformly generated independently for every $i = 0, \cdots, n-1$, then we finish the zero-knowledge proof of sTBoRP. □

4) Proof of Traceability:

Theorem 9 (Traceability of sTBoRP): sTBoRP is traceable for any PPT adversary $\mathcal{A}$ (without possession of trapdoors).

Proof: For a PPT adversary $\mathcal{A}$ without possession of the trapdoors, when $\mathcal{A}$ finished the tracing game with $\mathcal{S}$ in Definition 5, $\mathcal{A}$ generates a commitment c for a hidden amount a and range proof $\pi_{sTBo}(c) = (c, \beta, \{c_i, c_i'\}_{i \in [0,n-1]}, \{TK_{i,j}\}_{j \in [1,m]}^{i \in [0,n-1]}, \sigma)$, We assume that $\mathcal{A}$ wins the tracing game with nonnegligible advantage δ, that is, $\pi_{sTBo}(c)$ satisfying the following:

$$\mathsf{Verify}(\pi_{sTBo}(c)) = 1 \text{ and } \exists l \ s.t. \ \mathsf{Trace}(\pi_{sTBo}(c), y_l) \neq a.$$

According to the soundness of sTBoRP, we know $c = g^x h^a$ with $a \in [0, 2^n - 1]$ and $c_i = g^{x_i} h^{2^i a_i}$ for every $i = 0, \cdots, n-1$ except for negligible probability ϵ_1, we set $TK_{i,j} = g^{s_{i,j}} h^{v_{i,j}} \prod_{d=1}^m h_d^{t_{i,j,d}}$, then we get $e_k = H(c_0, \cdots, c_{n-1}, \{TK_{i,j}\}_{j \in [1,m]}^{i \in [0,n-1]}, k)$ for $k = 1, \cdots, m$ and

$$L_{PK}^i = \{c_i \cdot \prod_{j=1}^m TK_{i,j}^{e_j}, c_i' \cdot \prod_{j=1}^m TK_{i,j}^{e_j}\} = \{PK_i, PK_i'\}.$$

Without loss of generality we assume PK_i is the corresponding signing key, then

$$c_i \cdot \prod_{j=1}^m TK_{i,j}^{e_j} = g^{x_i + \sum_{j=1}^m s_{i,j} e_j} h^{2^i a_i + v_{i,j} e_j} \prod_{d=1}^m h_d^{\sum_{j=1}^m t_{i,j,d} e_j}$$
$$= (g \cdot \prod_{j=1}^m h_j^{e_j})^z = PK_i^z,$$

and z is known to $\mathcal{A}$ under the unforgeability of Borromean ring signature. We have $x_i - z + \sum_{j=1}^m s_{i,j} e_j = 0$, otherwise

$\mathcal{A}$ gets a non-trivial relation between $g, h, h_1, \cdots, h_m$, which happens with negligible probability ϵ_2, then we have $z = x_i$ and $s_{i,j} = 0$ (e_j cannot be pre-computed). From similar argument, we can also get $a_i = v_{i,j} = 0$, otherwise $\mathcal{A}$ gets a non-trivial relation between $g, h, h_1, \cdots, h_m$, which happens with negligible probability ϵ_3. For $\forall\ d \in [1, m]$, we know $h_d^{\sum_{j=1}^m t_{i,j,d} e_j} = h_d^{z e_d}$ from similar argument, except for negligible probability ϵ_4, then we have $z e_d = \sum_{j=1}^m t_{i,j,d} e_j$ and can get $t_{i,d,d} = z = x_i, t_{i,j,d} = 0$ for $d \neq j$, except for negligible probability ϵ_5 by the unpredictable of hash function, then we have $\mathsf{Trace}(\pi_{sTBo}(c), y_j) = a$ for $\forall\ j \in [1, m]$ with advantage $\delta - \sum \epsilon_i$, which contradicts with the assumptions before, then we get the traceability of sTBoRP. $\square$

C. Modification

The traceability of sTBoRP only holds for any PPT adversary $\mathcal{A}$ without possession of trapdoors, we can further modify sTBoRP to achieve traceability against malicious regulators by adding extra key-images to the proof, while maintaining the functionality of multiple regulation.

We introduce the modification of sTBoRP in the following:

$\mathsf{Par} \leftarrow \mathsf{Setup}(\lambda)$:

1. System chooses elliptic curve $\mathbb{G}$ with prime order q and samples independent generators $g, h \leftarrow \mathbb{G}$. For $j = 1, \cdots, m$, regulator $\mathcal{R}_j$ generates $y_j \in \mathbb{Z}_q^*$ as his trapdoor, computes $h_j = g^{y_j}$, system outputs $(\mathbb{G}, q, g, h, h_1, \cdots, h_m)$ as the public parameters.

$(c, \beta, SK, \{(c_i, c_i', I_i), TK_{i,j}\}_{j\in[1,m]}^{i\in[0,n-1]}) \leftarrow \mathsf{Gen}(\mathsf{Par}, a)$:

1. According to the public parameters and amount $a \in [0, 2^n - 1]$, prover Alice samples $x \in \mathbb{Z}_q^*$ uniformly, computes $c = g^x h^a$ as the commitment;
2. Alice computes the binary expansion $a = a_0 + \cdots + 2^{n-1} a_{n-1}, a_i = 0, 1$ for $i = 0, \cdots, n-1$, samples $x_0, \cdots, x_{n-1}$ uniformly, computes $\beta = x - x_0 - \cdots - x_{n-1}$;
3. For every $i = 0, \cdots, n-1$, Alice computes $c_i = g^{x_i} h^{2^i a_i}, c_i' = g^{x_i} h^{2^i a_i - 2^i}, I_i = h^{x_i}$, outputs $\{c_i, c_i', I_i\}_{i\in[0,n-1]}$;
4. For every $i = 0, \cdots, n-1$, $j = 1, \cdots, m$, Alice computes $TK_{i,j} = h_j^{x_i}$, then computes $e_k = H(\{c_i, I_i\}_{i\in[0,n-1]}, \{TK_{i,j}\}_{j\in[1,m]}^{i\in[0,n-1]}, k)$ for $k = 0, \cdots, m$;
5. Alice computes $L_{PK}^i = \{c_i I_i^{e_0} \cdot \prod_{j=1}^m TK_{i,j}^{e_j}, c_i' I_i^{e_0} \cdot \prod_{j=1}^m TK_{i,j}^{e_j}\}$ for $i = 0, \cdots, n-1$;
6. Alice outputs $(c, \beta, \{(c_i, c_i', I_i), TK_{i,j}\}_{j\in[1,m]}^{i\in[0,n-1]})$, retains $(a, SK = (x_0, \cdots, x_{n-1}), \{L_{PK}^i\}_{i\in[0,n-1]})$.

$\pi_{sTBo}(c) \leftarrow \mathsf{Prove}(SK, c, L_{PK})$:

1. For message μ, Alice runs the Borromean ring signature with $L_{PK} = \{L_{PK}^0, \cdots, L_{PK}^{n-1}\}$, computes $\sigma \leftarrow \mathsf{Rsign}(SK, \mu, L_{PK})$;
2. Alice outputs the sTBoRP proof $\pi_{sTBo}(c) = (c, \beta, \{c_i, c_i', I_i\}_{i\in[0,n-1]}, \{TK_{i,j}\}_{j\in[1,m]}^{i\in[0,n-1]}, \sigma)$.

$1/0 \leftarrow \mathsf{Verify}(\pi_{sTBo}(c))$:

1. For $k = 0, \cdots, m$, verifier computes $e_k = H(\{c_i, I_i\}_{i\in[0,n-1]}, \{TK_{i,j}\}_{j\in[1,m]}^{i\in[0,n-1]}, k)$;
2. Verifier computes $L_{PK}^i = \{c_i I_i^{e_0} \cdot \prod_{j=1}^m TK_{i,j}^{e_j}, c_i' I_i^{e_0} \cdot \prod_{j=1}^m TK_{i,j}^{e_j}\}$ for $i = 0, \cdots, n-1$;
3. For every $i = 0, \cdots, n-1$, verifier checks $c_i / c_i' \stackrel{?}{=} h^{2^i}$, then checks $g^\beta \cdot \prod_{i=0}^{n-1} c_i \stackrel{?}{=} c$;
4. Verifier checks the validity of Borromean ring signature σ, if all passed then outputs 1, otherwise outputs 0.

$a^* \leftarrow \mathsf{Trace}(y_j, \pi_{sTBo}(c))$, $j = 1, \cdots, m$:

1. For every $i = 0, \cdots, n-1$, $\mathcal{R}_j$ computes $c_i^{y_j}$;
2. For every $i = 0, \cdots, n-1$, if $c_i^{y_j} = TK_{i,j}$ then outputs $a_i^* = 0$, otherwise outputs $a_i^* = 1$;
3. $\mathcal{R}_j$ outputs $a^* = a_0^* + \cdots + 2^{n-1} a_{n-1}^*$ as the result.

Alg. 2. Modified sTBoRP

Note that $\{I_i\}_{i\in[0,n-1]}$ are the key-images of sTBoRP, and the Borromean ring signature Rsign (for n rings) uses $g h^{e_0} \cdot \prod_{j=1}^m h_j^{e_j}$ as the generator (basis element), correctness and security of modified sTBoRP easily follows from *Theorem 6-9*, the modified sTBoRP has traceability against malicious regulators, which means for any PPT adversary $\mathcal{A}$ who possesses all trapdoors, he cannot generate invalid proofs to escape from regulation, detailed proof will be given in the full version. Meanwhile, the message μ contains $c, \{c_i, c_i'\}_{i\in[0,n-1]}$ and other information related with the transaction, similar to Monero. Moreover, as in sTBoRP, all regulators can also recover the hidden amount by their trapdoors respectively. c_i' can also be removed from the proof since $c_i' = c_i / h^{2^i}$.

IV. New TBuRP for Joint Regulation

In this section we introduce jTBuRP, a modification of TBuRP to realize the functionality of joint regulation against collusion attack of malicious (branch) regulators, which means any number ($< m$) of corrupted branch regulators cannot recover any bit of the hidden amount, unless all of branch regulators collude.

A. Construction

The main modification of construction in jTBuRP is in the tracing key generation algorithm, we make merge operations of tracing keys which are related with trapdoor parameters to achieve secure joint regulation. To describe the jTBuRP scheme more clearly, we use a example of parameters in the construction, we set the amount bits to be $n = 32$, assume there are $m = 4$ branch regulators $\mathcal{R}_A, \mathcal{R}_B, \mathcal{R}_C, \mathcal{R}_D$, each regulator generates 2 trapdoors, each trapdoor is related with 4 bits, the regulation threshold is 4 (that is, for regulators with number less than 4, they cannot trace any bit of the amount). In our example, the amount bit partition $P = ((4,4)_A, (4,4)_B, (4,4)_C, (4,4)_D)$ is a uniform case, it should be noted that non-uniform distribution also applies to our scheme, we omit it due to its complicated expression. We set the chief regulator is $\mathcal{R}$. In fact, the number of amount bits, number of branch regulators, number of trapdoors, number of threshold are not restricted, anyone can change the parameter selection for different applications and regulatory policies.

We give the introduction of jTBuRP in the following:

$\mathsf{Par} \leftarrow \mathsf{Setup}(\lambda)$:

1. System chooses $\mathbb{G}$ with prime order q and generators $g, h, g_0, \cdots, g_{n-1} \in \mathbb{G}$, where $n = 32$;
2. $\mathcal{R}_A$ generates $y_0, y_1 \in \mathbb{Z}_q^*$, $\mathcal{R}_B$ generates $y_2, y_3 \in \mathbb{Z}_q^*$, $\mathcal{R}_C$ generates $y_4, y_5 \in \mathbb{Z}_q^*$, $\mathcal{R}_D$ generates $y_6, y_7 \in \mathbb{Z}_q^*$ as their trapdoors respectively;
3. All branch regulators compute $h_i = g_i^{y_{\lfloor i/4 \rfloor}}$ for $i = 0, \cdots, n-1$;
4. System outputs $(\mathbb{G}, q, g, h, \mathbf{g}, \mathbf{h}, P)$ as the public parameters, where $\mathbf{g} = (g_0, \cdots, g_{n-1}) \in \mathbb{G}^n$, $\mathbf{h} = (h_0, \cdots, h_{n-1}) \in \mathbb{G}^n$ and $P = ((4,4), (4,4), (4,4), (4,4))$ is the partition of amount bits.

$(A, S, c, \{TK_i\}_{i=0,\cdots,19}, \pi) \leftarrow \mathsf{Gen}(\mathsf{Par}, a)$:

1. According to the amount $a \in [0, 2^n - 1]$ and the public parameters $(\mathbb{G}, q, g, h, \mathbf{g}, \mathbf{h}, P)$, prover Alice samples $x \in \mathbb{Z}_q^*$ uniformly, computes $c = h^x g^a$ as the commitment;
2. Alice computes the binary expansion $a = a_0 + \cdots + 2^{n-1} a_{n-1}, a_i = 0, 1$ for $i = 0, \cdots, n-1$, sets $\mathbf{a}_L = (a_0, \cdots, a_{n-1})$, then computes $\mathbf{a}_R = \mathbf{a}_L - \mathbf{1}^n = (a_0 - 1, \cdots, a_{n-1} - 1)$;
3. Alice samples $\alpha \in \mathbb{Z}_q$ uniformly, then computes:
$$A = h^\alpha \mathbf{g}^{\mathbf{a}_L} \mathbf{h}^{\mathbf{a}_R} = h^\alpha g_0^{a_0} \cdots g_{n-1}^{a_{n-1}} h_0^{a_0 - 1} \cdots h_{n-1}^{a_{n-1}-1};$$
4. Alice samples $\mathbf{s}_L, \mathbf{s}_R \in \mathbb{Z}_q^n, \rho \in \mathbb{Z}_q$ uniformly at random, computes $S = h^\rho \mathbf{g}^{\mathbf{s}_L} \mathbf{h}^{\mathbf{s}_R}$;
5. For every $j = 0, \cdots, 15$, Alice computes $TK_j = g_{2j}^{\alpha - a_{2j}} g_{2j+1}^{\alpha - a_{2j+1}}$, then computes $TK_{16+k} = \prod_{i=0}^{3} (h_{8i+2k}^{-\alpha - a_{8i+2k}+1} h_{8i+2k+1}^{-\alpha - a_{8i+2k+1}+1})$ for $k = 0, \cdots, 3$, the total number of TK_is is 20;
6. Alice gives the validity proof $\pi(TK_0, \cdots, TK_{19}, A)$ for all TK_is, such that TK_j is a production of g_{2j}'s power and g_{2j+1}'s power for $j = 0, \cdots, 15$, TK_{16+k} is a product of $\{h_{8i+2k}, h_{8i+2k+1}\}_{i=0,\cdots,3}$'s power for $k = 0, \cdots, 3$, and $A \cdot \prod_{i=0}^{19} TK_i = (h \prod g_i / \prod h_i)^\alpha$ is a power of $h \prod g_i / \prod h_i$;
7. Alice outputs $(A, S, c, \{TK_i\}_{i=0,\cdots,19}, \pi)$, where $\pi = \pi(TK_0, \cdots, TK_{19}, A)$.

$(\pi, \pi', \{TK_i\}_{i=0,\cdots,19}) \leftarrow \mathsf{Prove}(A, S, c, a, v)$:

1. Prover sends $(A, S, c, \{TK_i\}_{i=0,\cdots,19}, \pi$ to verifier;
2. Verifier samples $y, z \in \mathbb{Z}_q$ uniformly at random, and sends them to prover;
3. Prover computes T_1, T_2 and sends them to verifier;
4. Verifier samples $v \in \mathbb{Z}_q$ uniformly at random, and sends it to prover;
5. Prover computes $\tau_v, \mu, t, \mathbf{l}, \mathbf{r}$, sends them to verifier;
6. Prover outputs $\pi_{jTBu} = (\{TK_i\}_{i=0,\cdots,19}, \pi, \pi')$ as the jTBuRP result, where $\pi' = (T_1, T_2, \tau_v, \mu, t, \mathbf{l}, \mathbf{r})$.

$1/0 \leftarrow \mathsf{Verify}(\pi_{jTBu}(c))$: we only introduce the verification of $\pi = \pi(TK_0, \cdots, TK_{19}, A)$:

1. For every $i = 0, \cdots, 19$, verifier checks the validity of TK_i;
2. Verifier computes $A \cdot \prod_{i=0}^{19} TK_i$ and checks the validity of $A \cdot \prod_{i=0}^{19} TK_i$;
3. Verifier checks the validity of Bulletproofs result π';
4. If all passed then outputs 1, otherwise outputs 0.

$a^* \leftarrow \mathsf{Trace}(\{TK_i\}_{i=0,\cdots,19}, y_0, \cdots, y_7)$:

1. For every $j = 0, \cdots, 15$, all branch regulators compute $T_j = TK_j^{y_{\lfloor j/2 \rfloor}}$ and send them to the chief regulator $\mathcal{R}$;
2. $\mathcal{R}$ searches for $d_i \in \{-1, 1\}$ such that $TK_{16} \cdot T_0 T_4 T_8 T_{12} = h_0^{d_0} h_1^{d_1} h_8^{d_8} h_9^{d_9} h_{16}^{d_{16}} h_{17}^{d_{17}} h_{24}^{d_{24}} h_{25}^{d_{25}}$, outputs $a_i^* = \frac{1}{2} - \frac{1}{2} d_i$ for $i = 0, 1, 8, 9, 16, 17, 24, 25$;
3. $\mathcal{R}$ searches for $d_i \in \{-1, 1\}$ such that $TK_{17} \cdot T_1 T_5 T_9 T_{13} = h_2^{d_2} h_3^{d_3} h_{10}^{d_{10}} h_{11}^{d_{11}} h_{18}^{d_{18}} h_{19}^{d_{19}} h_{26}^{d_{26}} h_{27}^{d_{27}}$, outputs $a_i^* = \frac{1}{2} - \frac{1}{2} d_i$ for $i = 2, 3, 10, 11, 18, 19, 26, 27$;
4. $\mathcal{R}$ searches for $d_i \in \{-1, 1\}$ such that $TK_{18} \cdot T_2 T_6 T_{10} T_{14} = h_4^{d_4} h_5^{d_5} h_{12}^{d_{12}} h_{13}^{d_{13}} h_{20}^{d_{20}} h_{21}^{d_{21}} h_{28}^{d_{28}} h_{29}^{d_{29}}$, outputs $a_i^* = \frac{1}{2} - \frac{1}{2} d_i$ for $i = 4, 5, 12, 13, 20, 21, 28, 29$;
5. $\mathcal{R}$ searches for $d_i \in \{-1, 1\}$ such that $TK_{19} \cdot T_3 T_7 T_{11} T_{15} = h_6^{d_6} h_7^{d_7} h_{14}^{d_{14}} h_{15}^{d_{15}} h_{22}^{d_{22}} h_{23}^{d_{23}} h_{30}^{d_{30}} h_{31}^{d_{31}}$, outputs $a_i^* = \frac{1}{2} - \frac{1}{2} d_i$ for $i = 6, 7, 14, 15, 22, 23, 30, 31$;
6. The chief regulator $\mathcal{R}$ outputs $a^* = a_0^* + \cdots + 2^{n-1} a_{n-1}^*$ as the tracing result.

Alg. 3. jTBuRP

It should be emphasized that the above scheme is an interactive zero-knowledge proof system, we can easily turn it into a non-interactive scheme by usage of Fiat-Shamir transformation. The validity proof of tracing keys $\pi = \pi(TK_0, \cdots, TK_{19}, A)$ is derived from the proof of committed values, which is given in II.A. Moreover, the chief regulator $\mathcal{R}$ can be an independent regulator with no trapdoors, who only receives messages T_j from the branch regulators, he also can be one of the branch regulators, it is optional according to the actual scenario. Another option of jTBuRP is to remove the role of chief regulator that all branch regulators can recover the hidden amount by MPC, which is the closest approach to "decentralization". All solutions are technically feasible, we do not make a choice in this paper and leave it in the future work.

B. Correctness and Security

Completeness of jTBuRP follows from the correctness, soundness of jTBuRP follows from the soundness of Bulletproofs [9], zero-knowledge and traceability of jTBuRP only holds for PPT adversary $\mathcal{A}$ without possession of the trapdoors.

1) Proof of Correctness:

Theorem 10 (Correctness of jTBuRP): The hidden amount a in jTBuRP scheme can be correctly traced by regulators jointly.

Proof: $T_j = TK_j^{y_{\lfloor j/2 \rfloor}} = g_{2j}^{(\alpha - a_{2j}) y_{\lfloor j/2 \rfloor}} g_{2j+1}^{(\alpha - a_{2j+1}) y_{\lfloor j/2 \rfloor}} = h_{2j}^{\alpha - a_{2j}} h_{2j+1}^{\alpha - a_{2j+1}}$, for $j = 0, \cdots, 15$, then we have for $k = 0, 1, 2, 3$:

$$TK_{16+k} \cdot \prod_{i=0}^{3} T_{4i+k}$$

$$= \prod_{i=0}^{3} (h_{8i+2k}^{-\alpha - a_{8i+2k}+1+\alpha - a_{8i+2k}} h_{8i+2k+1}^{-\alpha - a_{8i+2k+1}+1+\alpha - a_{8i+2k+1}})$$

$$= \prod_{i=0}^{3} (h_{8i+2k}^{-2a_{8i+2k}+1} h_{8i+2k+1}^{-2a_{8i+2k+1}+1}) = \prod_{i=0}^{3} (h_{8i+2k}^{d_{8i+2k}} h_{8i+2k+1}^{d_{8i+2k+1}}).$$

Then we get $a_j = \frac{1}{2} - \frac{1}{2} d_j \in \{0, 1\}$ for $j = 0, \cdots, n-1$, and we get the correctness of jTBuRP. □

2) Proof of Zero-knowledge:

Theorem 11 (Zero-knowledge of jTBuRP): The jTBuRP scheme has zero-knowledge for any PPT adversary $\mathcal{A}$ without possession of trapdoors.

Proof: Since $TK_j = g_{2j}^{\alpha-a_{2j}} g_{2j+1}^{\alpha-a_{2j+1}}$, $j = 0, \cdots, 15$, and $TK_{16+k} = \prod_{i=0}^{3}(h_{8i+2k}^{-\alpha-a_{8i+2k}+1} h_{8i+2k+1}^{-\alpha-a_{8i+2k+1}+1})$, $k = 0, \cdots, 3$. Similar to [10], we can prove when $S_i = g_i^{\alpha-a_i}, V_i = h_i^{\alpha+a_i} (i = 0, \cdots, n-1)$ is added into the jTBuRP proof π_{jTBu} instead of $\{TK_j\}_{j\in[0,19]}$, jTBuRP maintains the zero-knowledge property.

As $\{TK_j\}_{j\in[0,19]}$ can be computed easily from $\{S_i, V_i\}_{i\in[0,n-1]}$, then we know that $\{S_i, V_i\}_{i\in[0,n-1]}$ contains more informations than $\{TK_j\}_{j\in[0,19]}$, so if adding $\{S_i, V_i\}_{i\in[0,n-1]}$ to Bulletproofs has no effects on the zero-knowledge property, then we can get jTBuRP also has zero-knowledge, under the DDH assumption.

As shown in [10], they have already proved that the extra $\{S_i, V_i\}_{i\in[0,n-1]}$ have no effect on zero-knowledge of Bulletproofs, then we get the zero-knowledge of jTBuRP. □

3) Proof of Traceability:

Theorem 12 (Traceability of jTBuRP): The jTBuRP scheme has traceability for any PPT adversary $\mathcal{A}$ without possession of trapdoors.

Proof: The traceability of jTBuRP easily follows from the soundness of validity proof for tracing keys $\pi = \pi(TK_0, \cdots, TK_{19}, A)$, which is the proof of committed values from II.A. Then we know $\pi(TK_0, \cdots, TK_{19}, A)$ has soundness for any PPT adversary $\mathcal{A}$ without possession of trapdoors, so $\mathcal{A}$ cannot generate invalid tracing keys to escape from regulation, then we get the traceability of jTBuRP. □

4) Threshold Analysis: Consider the regulatory threshold of jTBuRP, that is, any number (less than the threshold) of branch regulators can not recover any bit of the hidden amount from jTBuRP, this strengthen the security of jTBuRP when regulators are attacked, corrupted or collusive. In the jTBuRP example with partition $P = ((4,4)_A, (4,4)_B, (4,4)_C, (4,4)_D)$, we can show that any number (< 4) of branch regulators cannot do better than guessing the hidden amount.

Theorem 13 (Security against collusion attack): The regulatory threshold of jTBuRP in section IV.A is 4, the hidden amount will be leaked iff all regulators collude. In other words, jTBuRP is zero-knowledge for any PPT adversary $\mathcal{A}$ (not corrupt all regulators).

Proof: Without loss of generality, we assume a PPT adversary $\mathcal{A}$ who corrupts $\mathcal{R}_B, \mathcal{R}_C, \mathcal{R}_D$ (possessing their trapdoors $y_2 \cdots, y_7$), but does not know the trapdoors (y_0, y_1) of $\mathcal{R}_A$.

In $\mathcal{A}$'s view, we prove that for any $a \in [0, 2^n - 1]$ with the corresponding $c = h^x g^a$, $\pi_{jTBu}(c)$ is distinguishable from uniform distribution. Actually, we assume $TK_j = g_{2j}^{\alpha-a_{2j}} g_{2j+1}^{\alpha-a_{2j+1}}$ for $j = 0, \cdots, 15$ is replaced with $(S_{2j}, S_{2j+1}) = (g_{2j}^{\alpha-a_{2j}}, g_{2j+1}^{\alpha-a_{2j+1}})$, and $TK_{16+k} = \prod_{i=0}^{3}(h_{8i+2k}^{-\alpha-a_{8i+2k}+1} h_{8i+2k+1}^{-\alpha-a_{8i+2k+1}+1})$ for $k = 0, \cdots, 3$ is replaced with $V_{2k} = \prod_{i=0}^{3} h_{8i+2k}^{-\alpha-a_{8i+2k}+1}$ and $V_{2k+1} = \prod_{i=0}^{3} h_{8i+2k+1}^{-\alpha-a_{8i+2k+1}+1}$, then the extra information in $\pi_{jTBu}(c)$ is $(\{S_j\}_{j\in[0,31]}, \{V_k\}_{k\in[0,7]})$. We only need to prove that $(\{S_j\}_{j\in[0,31]}, \{V_k\}_{k\in[0,7]})$ is computational indistinguishable from uniform distribution $U \leftarrow \mathbb{G}^{40}$.

From similar argument in *Theorem 11* we know that $\{S_j\}_{j\in[0,31]}$ is computational indistinguishable from uniform distribution $U_0 \leftarrow \mathbb{G}^{32}$. Since $h_i = g_i^{y_{\lfloor i/4 \rfloor}}$ for $i = 0, \cdots, 7$ with the corresponding trapdoors y_0, y_1 not possessed by $\mathcal{A}$, we have $\{h_k^{-\alpha-a_k+1}\}_{k=0,\cdots,7}$ is computational indistinguishable from uniform distribution $U_1 \leftarrow \mathbb{G}^8$, then we have $\{V_k\}_{k\in[0,7]}$ is also computational indistinguishable from uniform distribution $U_1 \leftarrow \mathbb{G}^8$, we get $(\{S_j\}_{j\in[0,31]}, \{V_k\}_{k\in[0,7]})$ is computational indistinguishable from uniform distribution $U \leftarrow \mathbb{G}^{40}$ for any $a \in [0, 2^n - 1]$. Then for we know jTBuRP is zero-knowledge for any PPT adversary $\mathcal{A}$ who does not corrupt all regulators, and get the regulatory threshold of jTBuRP is 4. □

V. Analysis and Comparison

1) sTBoRP vs. *TBoRP* vs. *BoRP*: We give a brief comparison of efficiency between sTBoRP($m = 1$), TBoRP and BoRP in Table1, where we compare the sizes of proofs, computations of generation and verification in each scheme as well as the security between them, where size $(\#\mathbb{G}, \#\mathbb{Z}_q^*)$ is denoted by number of elements in $(\mathbb{G}, \mathbb{Z}_q^*)$, computations of generation and verification $(\#\text{exponentiation}, \#\text{multiplication})$ are denoted by number of exponentiations and multiplications in $\mathbb{G}$, while computations of Hash and computations in $\mathbb{Z}_q^*$ is ignored. Soundness and traceability refer to the corresponding adversary type, n refers to the maximum bit length of the hidden amount. From Table1 we know that sTBoRP (sTBoRP') is more efficient than TBoRP (TBoRP' [10]) in size, generation time and verification time with enhanced security.

Scheme	Size	Generation	Verification	Soundness	Traceability
sTBoRP	($2n$, $2n$+2)	($6n$+1, $3n$+2)	($5n$+2, $6n$+1)	Malicious	Honest
sTBoRP'	($3n$, $2n$+2)	($8n$+2, $4n$+3)	($6n$+3, $7n$+2)	Malicious	Malicious
TBoRP	($2n$, $2n$+5)	($7n$+2, $4n$-1)	($6n$+5, $6n$+1)	Honest	Honest
TBoRP'	($3n$+1, $2n$+7)	($8n$+3, $5n$-1)	($7n$+7, $8n$+2)	Malicious	Malicious
BoRP	(n, $2n$+2)	($4n$, $2n$)	($4n$+1, $4n$)	Malicious	None

Table1. *sTBoRP* vs. *TBoRP* vs. *BoRP*

2) jTBuRP vs. *TBuRP* vs. *BuRP*: We give a brief comparison between jTBuRP, TBuRP and BuRP in Table2, where we focus on the comparison of the **extra** computations (#exponentiation,#multiplication) other than the computations in original Bulletproofs range proof (BuRP) in generation and verification between the schemes. Moreover, in BuRP, the original size (number of elements in $(\mathbb{G}, \mathbb{Z}_q^*)$) is $(2\lceil \log n \rceil + 4, 5)$, and the number of computations for generation is $O(n)$ and verification is $O(n/\log n)$. We assume $n = 32$, then $(2\lceil \log n \rceil + 4, 5) = (14, 5)$. From Table2 we know that jTBuRP is more efficient than TBuRP in size, generation time and verification time.

Scheme	Size	Extra Gen	Extra Ver	Soundness	Traceability
jTBuRP	(34, 71)	(129, 88)	(86, 97)	Malicious	Honest
TBuRP	(46, 71)	(129, 64)	(98, 97)	Malicious	Honest
BuRP	(14,5)	(0, 0)	(0, 0)	Malicious	None

Table2. *jTBuRP* vs. *TBuRP* vs. *BuRP*

VI. Conclusion

In this paper, we introduce new constructions of traceable range proofs, including simplified traceable Borromean range proof (sTBoRP) and its application in multiple regulation, modified traceable Bulletproofs range proof (jTBuRP) and its application in joint regulation. For sTBoRP, it can be further modified to be secure against malicious regulators. For jTBuRP, it has security against collusion attack between malicious (branch) regulators.

References

[1] S. Nakamoto *et al.*, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, p. 37, 2014.

[3] N. Van Saberhagen, "Cryptonote v 2.0," *URL: https://cryptonote.org/whitepaper.pdf*, pp. 04–13, 2013.

[4] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE Symposium on Security and Privacy*. IEEE, 2014, pp. 459–474.

[5] G. Maxwell, "Confidential transactions," 2015.

[6] T. E. Jedusor, "Mimblewimble (2016)," *Defunct hidden service*, 2017.

[7] E. Duffield and D. Diaz, "Dash: A privacycentric cryptocurrency," *No Publisher*, 2015.

[8] S. Noether, A. Mackenzie *et al.*, "Ring confidential transactions," *Ledger*, vol. 1, pp. 1–18, 2016.

[9] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 315–334.

[10] W. Li, L. Chen, X. Lai, X. Zhang, and J. Xin, "Fully regulatable privacy-preserving blockchains against malicious regulators," Cryptology ePrint Archive, Report 2019/925, 2019.

[11] G. Maxwell and A. Poelstra, "Borromean ring signatures," 2015.

[12] D. Community, "Dero white paper." *URL: https://dero.io/attachment/Whitepaper.pdf*, 2019.

[13] P. Fauzi, S. Meiklejohn, R. Mercer, and C. Orlandi, "Quisquis: A new design for anonymous cryptocurrencies." *IACR Cryptology ePrint Archive*, vol. 2018, p. 990, 2018.

[14] B. Bünz, S. Agrawal, M. Zamani, and D. Boneh, "Zether: Towards privacy in a smart contract world." *IACR Cryptology ePrint Archive*, vol. 2019, p. 191, 2019.

[15] R. S. Wahby, I. Tzialla, A. Shelat, J. Thaler, and M. Walfish, "Doubly-efficient zksnarks without trusted setup," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 926–943.

[16] T. Xie, J. Zhang, Y. Zhang, C. Papamanthou, and D. Song, "Libra: Succinct zero-knowledge proofs with optimal prover computation." *IACR Cryptology ePrint Archive*, vol. 2019, p. 317, 2019.

[17] H. Wu, W. Zheng, A. Chiesa, R. A. Popa, and I. Stoica, "Dizk: A distributed zero knowledge proof system," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 675–692.

[18] J. Groth, M. Kohlweiss, M. Maller, S. Meiklejohn, and I. Miers, "Updatable and universal common reference strings with applications to zk-snarks," in *Annual International Cryptology Conference*. Springer, 2018, pp. 698–728.

[19] E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward, "Aurora: Transparent succinct arguments for r1cs," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2019, pp. 103–128.

[20] B. Bünz, B. Fisch, and A. Szepieniec, "Transparent snarks from dark compilers," 2019.

[21] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Annual International Cryptology Conference*. Springer, 1991, pp. 129–140.