Privacy-Preserving Public Verification of Ethical Cobalt Sourcing

Kilian Becher
Chair of Privacy and Data Security
TU Dresden
Dresden, Germany
kilian.becher@tu-dresden.de

J. A. Gregor Lagodzinski
Hasso Plattner Institute
University of Potsdam
Potsdam, Germany
gregor.lagodzinski@hpi.de

Thorsten Strufe
Chair of IT Security
Karlsruhe Institute of Technology
Karlsruhe, Germany
thorsten.strufe@kit.edu

***Abstract*—Cobalt is a key ingredient of lithium-ion batteries and therefore is crucial for many modern devices. To ensure ethical sourcing, consumers need a way to verify provenance of their cobalt-based products, including the percentage of artisanally mined (ASM) cobalt. Existing frameworks for provenance and supply chain traceability rely on distributed ledgers. Providing public verifiability via permissionless distributed ledgers is trivial. However, offering public verifiability based on confidential production details seems contradictory. Hence, existing frameworks lack public verifiability of ratios between commodities while ensuring confidentiality of supply chain details.**

We propose a protocol that allows end consumers to verify the percentage of ASM cobalt in their products. Unlike previous solutions, production details are published and processed entirely in encrypted form by employing homomorphic encryption and proxy re-encryption. Thus, it ensures a high level of confidentiality of supply chain data. It has constant consumer-side complexity, making it suitable for mobile devices.

***Index Terms*—Homomorphic encryption, proxy re-encryption, distributed ledger technology, cobalt provenance**

I. Introduction

The chemical element cobalt is a main ingredient for modern lithium-ion batteries, such as used for mobile phones, notebooks, and electric cars [22]. With a growing demand for lithium-ion batteries, the demand for cobalt will likely grow proportionately [4]. This demand is met mostly by the Democratic Republic of the Congo (DRC), which accounts for more than $50\,\%$ of the world's cobalt reserves and more than $70\,\%$ of the annual global cobalt production of approximately $140,000$ tons [22]. The majority of this cobalt is mined by large-scale mining (LSM) companies with heavy machinery. However, an estimated $20\,\%$ of the DRC's cobalt is mined by more than $100,000$ artisanal and small-scale miners (ASM) [4].

ASMs often mine with hand tools and only little protection and safety measures. As revealed by Amnesty International [4], artisanal miners in the DRC face health risks such as back injury and lung diseases [8, 27] as well as accidents due to collapsing tunnels or underground fires [17]. Reportedly, artisanal mining frequently involves child labor [4]. However, with more than $70\,\%$ of the DRC's population living in extreme poverty [24], artisanal mining secures the livelihood of many people. Hence, simply prohibiting ASM activities and excluding artisanal mines from the cobalt supply chain cannot be considered optimal from an ethical perspective as it might drive many people deeper into poverty [25]. Instead, to improve and control the labor conditions of artisanal miners, the DRC started opening government-operated artisanal mining zones, "zones d'exploration artisanale" (ZEA) [4]. These official mines allow artisanal mining where industrial mining is not feasible and aim to ensure ethical labor conditions.

With ZEAs, some degree of ASM cobalt in a product can be acceptable, allowing products to come with claims like "$100\,\%$ ethical cobalt containing $20\,\%$ ASM cobalt from ZEAs." The verification of such claims requires provenance tracking from the product through the supply chain back to the cobalt mine. We assume that public verifiability, i.e., verifiability for any party including end consumers, increases public awareness and adds pressure to source ethically. Existing supply chain traceability and provenance verification frameworks, such as [2, 7, 16, 26], achieve traceability by relying on distributed ledgers.

To verify metrics like the percentage of ASM cobalt in a product, one needs to compute an arithmetic function on supply chain details. Combining public verifiability with confidentiality of those details is anything but trivial. Confidentiality ensures supply chain actors' competitive advantages. Hence, a solution's adoption depends on its confidentiality guarantees. To offer public verifiability, existing solutions sacrifice confidentiality of supply chain details. Solutions with sufficient confidentiality typically lack public verifiability. Hence, they do not allow end consumers to verify ratios between commodities and at the same time ensure a high level of confidentiality of supply chain details.

We propose a cryptographic protocol that allows end consumers to publicly verify the claimed percentage of ASM cobalt used to manufacture a product. This verification also takes ratios between different lots of cobalt ore into consideration and thus incorporates mixing of ingredients in the various supply chain steps. By combining the cryptographic techniques of fully homomorphic encryption and proxy re-encryption, our protocol ensures a high level of confidentiality of supply chain details. It ensures good scalability and requires very little computation and communication on the consumer side. With its ledger-agnostic design, our protocol augments existing distributed-ledger-based supply chain traceability systems by enabling verification on a product level by end consumers.

The paper is organized as follows. We first provide details

978-1-6654-0392-4/20/$31.00 ©2020 IEEE
DOI 10.1109/TrustCom50675.2020.00133

Fig. 1: Condensed Cobalt Supply Chain

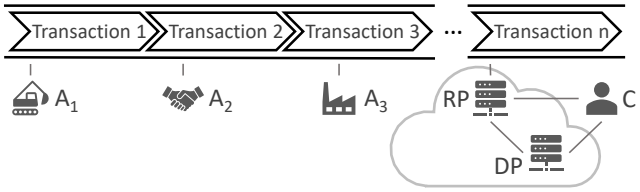


Fig. 2: Scenario with Three Supply Chain Actors A_1, A_2, A_3, One Consumer C, and Two Cloud Services RP, DP

on the cobalt scenario, briefly describe the concept of our solution, and discuss related work in Section II. Then, we introduce preliminaries in Section III. The verification protocol is described in detail in Section IV and evaluated in Section V.

II. Background and Related Work

A. Scenario Description and Concept

Figure 1 depicts a condensed illustration of the cobalt supply chain from mined cobalt ore to refined cobalt, to electronic devices. The supply chain could in reality contain more trading and manufacturing steps and typically consists of twelve supply chain stages [4, 14].

Independent of distributed-ledger-based supply chain traceability solutions, the OECD Due Diligence Guidance [18] requires on-ground assessment of supply chain actors by third-party auditors to ensure a minimum of ethical sourcing and human rights in the supply chain. We do not aim to replace these audits but rather complement them with end consumer verification in a privacy-preserving form. This verification can for example be triggered with a consumer's phone by scanning a QR code that is printed on the purchased product.

We target scenarios that provide every end consumer with the option to verify ethical cobalt sourcing but assume that in reality, given the above mentioned audits, only a fraction of end consumers will use this option. Hence, we aim to minimize supply chain actors' effort at transaction time and design our solution to verify ethical sourcing on-demand. This on-demand approach further allows verifiers to determine their definition of ethical sourcing on a product level at verification time.

We focus on enabling verification of the ratio between cobalt from large-scale mines (LSM) and artisanal and small-scale mines (ASM) in a product. That is, we verify the percentage of ASM cobalt. For this verification, we require confidentiality of the mined amounts of cobalt ore. Protecting the amounts also protects supply chain actors' trade secrets such as bills of materials of their products as well as storage capacities and warehouse stocks. Leaking this kind of information could put the supply chain actors' competitive advantages at risk.

For privacy-preserving ASM percentage computation, we need to compute an arithmetic function on confidential information. We perform this computation such that no one learns the confidential supply chain details that are used to compute the ASM percentage. Furthermore, we require that no one except for the verifying consumer learns the output, i.e., ASM percentage. Performing this computation is anything but trivial and involves sophisticated cryptographic techniques.

We base our privacy-preserving verification protocol on a distributed ledger that reflects all supply chain transactions, as in existing supply chain traceability solutions. Certified supply chain actors, e.g., miners, traders, or manufacturers, write distributed ledger entries containing production details of the respective supply chain stages. These details may comprise valuable information such as actor identities, asset types, and locations. Miners' distributed ledger entries carry the mined amount of cobalt ore in encrypted form and also indicate whether it was mined by an LSM or an ASM. This information can be implicitly based on the miner's identity contained in the respective distributed ledger entry. Anyone with access to the distributed ledger can read these details and evaluate claims made about their purchased products.

Each distributed ledger entry indicates parent-child relations such that one can traverse the distributed ledger from a given end product all the way back to the mining entries that led to the cobalt used in the product. This allows verifying the cobalt origin and therefore allows provenance verification as in existing supply chain traceability solutions (see Section I).

Our protocol relies on an architecture that involves two central, neutral parties: a re-encryption party RP and a decryption party DP (see Figure 2). In a dialog-like manner, the consumer C and the independent parties RP and DP jointly verify the claimed ASM percentage. They traverse the distributed ledger entries that have previously been written by supply chain actors A_i and read the encrypted mined amounts of LSM and ASM cobalt ore that led to the cobalt which was used to manufacture the product. These ciphertexts are re-encrypted under the same key and then used to compute the toal ASM and LSM amounts and eventually compute the ASM percentage as $\frac{\sum ASM}{\sum ASM + \sum LSM}$. Employing the re-encryption party RP and the decryption party DP allows us to encrypt the mined amounts under individual keys owned by the supply chain actors. These individual keys prevent unauthorized supply chain actors from learning other parties' private amounts. Homomorphic encryption allows us to process mined amounts entirely in encrypted form and therefore guarantees a high level of confidentiality throughout the verification of the ASM percentage. The neutral parties RP and DP could be hosted and controlled by NGOs, such as Amnesty International, which aim to improve labor conditions and achieve ethical sourcing of cobalt [4]. Hence, they can be assumed to have an intrinsic motivation to disclose misconduct in the supply chain if there is any indication.

B. Adversary Model

The proposed structure for distributed ledger entries that reflect supply chain actions allows for tracing the origin of

mined cobalt ore. Our protocol computes the percentage of ASM cobalt ore relatively to the total amount of cobalt ore used to manufacture a product. These amounts are published and processed entirely in encrypted form.

We consider an adversary that tries to learn single amounts or their sums in the face of public verifiability. It can corrupt either the requesting end consumer, the re-encryption party, or the decryption party. We assume semi-honest [15] consumers, re-encryption party, and decryption party and exclude collusion between the requesting consumer, re-encryption party, and decryption party. This non-collusion assumption is reasonable as long as the re-encryption party and the decryption party are controlled by independent entities that have an intrinsic motivation to disclose misconduct by supply chain actors if there is any indication, e.g., NGOs [4].

Cheating supply chain actors that write inconsistent or incorrect data onto the ledger are out of the scope of this work as the actors themselves are not involved in our verification. Misbehaving supply chain actors are not specific to our protocol and one of the main reasons to employ supply chain traceability systems. In the cobalt supply chain, actor-side fraud is tackled via third-party on-ground assessments (see Section II-A).

Direct communication between any pair of involved parties is assumed to be performed over pairwise secure and authentic channels, e.g., established via Transport Layer Security (TLS).

C. Related Work

1) Zero-Knowledge Proofs: We require our solution to allow verification of the ratio between different commodities or sources of raw materials in a privacy-preserving form. Privacy-preserving ratio verification can be based on zero-knowledge proofs (ZKP). ZKPs are a cryptographic technique that enables a prover P to convince a verifier V that P knows some secret x without revealing anything about x to V apart from the fact that x is known to P [12].

For ratio verification, the supply chain actors could prove in every production step the claimed ratio with a ZKP. However, this implies that either the supply chain actor actively participates in the verification in the form of an interactive proof or it has to prepare a non-interactive proof in advance for every transaction. The former only suits non-volatile supply chains as it precludes verification if the respective supply chain actor dropped out. The latter adds additional computational overhead and requires space on the distributed ledger for storing large proofs, even for those transactions that might never be verified by consumers. This large overhead might not be reasonable if only a small fraction of these proofs are ever going to be verified. Therefore, zero-knowledge proofs do not lend themselves very well to the described scenario.

2) Distributed-Ledger-Based Supply Chain Traceability: A plethora of supply chain traceability and provenance verification frameworks and systems have been proposed over the past few years. Many of them achieve traceability as well as transparency and integrity by reflecting supply chain transactions on a distributed ledger.

AgriBlockIoT [7] is a distributed-ledger-based, decentralized supply chain traceability system for the agriculture and food supply chain. Even though it provides users with a full history of the purchased food, it does not employ mechanisms that protect supply chain actors confidential transaction details.

ProductChain [16] is a distributed-ledger-based, permissioned framework for provenance in food supply chains. Consumers query a global validator to retrieve provenance information. This validator traverses the distributed ledger, reads provenance information, and provides necessary information to the consumer. This data contains provenance details such as farms that provided initial ingredients. However, the validator of ProductChain does not offer ratio verification for similar ingredients from different sources, e.g., farms.

A traceability system for the textile and clothing industry is presented in [2]. It employs the distributed ledger technology to track textile products through the supply chain, identify suppliers, and recognize counterfeits. Products are identified via forgery-proof tags. The system relies on using a permissioned distributed ledger but does not provide any further immanent privacy mechanisms.

A system for tracing the transformation of goods during their flow through the supply chain is presented in [26]. These goods are represented as tokens and their transformation is described in the form of recipes. Smart contracts perform transformations of tokens into new tokens according to these recipes. Even though this construction could help to track the transformation from cobalt ore to cobalt to components and end products, it does not take confidentiality of transaction details into account. Hence, it is not suitable for scenarios with highly confidential supply chain data.

Other solutions focus on provenance of data in collaboration scenarios rather than the flow of products and commodities through their supply chains. They have likewise objectives and use similar technologies. Ancile [9] is a distributed-ledger-based framework for managing medical records of patients. It gives control to patients and allows them to keep track of who is using their data. Confidential data is encrypted. The respective decryption keys can be transferred over the distributed ledger using a distributed, blinded re-encryption scheme that involves multiple proxies. Ancile does not offer privacy-preserving ratio computation.

III. Preliminaries

A. Homomorphic Encryption

We define asymmetric encryption schemes as tuples $\mathcal{S} = (G, E, D)$ with key-generation algorithm $G(\cdot)$, encryption algorithm $E(\cdot)$, and decryption algorithm $D(\cdot)$. $G(\cdot)$ generates a pair (pk, sk) of a public encryption and a secret decryption key. Encryption of a plaintext $m \in \mathcal{M}$ with pk yields the ciphertext $c = E_{pk}(m) \in \mathcal{C}$, were $\mathcal{M}$ and $\mathcal{C}$ are the plaintext and ciphertext space, respectively. Decryption of c with sk yields $m = D_{sk}(c)$.

Homomorphic encryption (HE) schemes additionally allow computations on ciphertexts that map to homomorphic operations on the underlying plaintexts. Assume two ciphertexts

$E_{pk}(m_1)$ and $E_{pk}(m_2)$ encrypted under the same public key pk of an (asymmetric) encryption scheme $\mathcal{S}$. $\mathcal{S}$ is homomorphic if it provides an operation "$\circ$" on $\mathcal{C}$ that corresponds to an operation "$\bullet$" on $\mathcal{M}$ such that $E_{pk}(m_1) \circ E_{pk}(m_2)$ yields an encryption of the plaintext operation $m_1 \bullet m_2$. For probabilistic encryption functions $E(\cdot)$, equivalence applies only on plaintext level. Hence, we denote homomorphic operations by

$$D_{sk}(E_{pk}(m_1) \circ E_{pk}(m_2)) = m_1 \bullet m_2. \quad (1)$$

Typically, those homomorphic operations are addition (see Equation (2)) or multiplication (see Equation (3)).

$$D_{sk}(E_{pk}(m_1) \oplus E_{pk}(m_2)) = m_1 + m_2 \quad (2)$$

$$D_{sk}(E_{pk}(m_1) \odot E_{pk}(m_2)) = m_1 \cdot m_2 \quad (3)$$

Some HE schemes allow ciphertext-plaintext operations, which combine an encrypted secret value with a known plaintext value and yield the encrypted result. These can be more efficient than pure ciphertext-ciphertext operations [6].

Partially homomorphic encryption (PHE) schemes, such as Paillier's [19] and RSA [21], enable homomorphic addition or multiplication. Fully homomorphic encryption (FHE) schemes, like BFV [10], allow both homomorphic addition and multiplication of encrypted secrets. Hence, the latter enable privacy-preserving evaluation of arbitrary arithmetic functions, however, typically with a high computational overhead [1].

Throughout this paper, we use short notations and denote encryption with a party P's public key pk_P by $c = E_P(m)$ and decryption with P's secret key sk_P by $m = D_P(c)$.

B. Proxy Re-Encryption

Re-encryption transforms ciphertexts encrypted under one key into ciphertexts of the same plaintext encrypted under a different key. In proxy re-encryption (PRE) [5], this transformation can be performed by an untrusted party without affecting confidentiality. A default way to implement PRE based on FHE is described in Gentry's seminal work [11].

Following the notation of [20], we define PRE schemes as tuples $\mathcal{PRE} = (PG, KG, ReKG, E, D, RE)$ of six procedures. The parameter generation algorithm $PG(\cdot)$ generates a set of public parameters. Given these parameters, the key generation procedure $KG(\cdot)$ outputs a key pair (pk, sk). The re-encryption key generation algorithm $ReKG(\cdot)$ takes the secret key sk_i as well as a public key $pk_{j \neq i}$ and outputs a re-encryption key $rk_{i \to j}$. The re-encryption function $RE(\cdot)$ takes a ciphertext $c_i = E_i(m)$ encrypted under pk_i together with $rk_{i \to j}$ and outputs a ciphertext $c_j = E_j(m)$ of m encrypted under pk_j. $E(\cdot)$ and $D(\cdot)$ are the encryption and decryption functions as defined in Section III-A.

C. Distributed Ledger

A distributed ledger is an append-only data storage that is maintained in a network of distributed nodes [13]. Data is organized in the form of transactions. When a transaction is added, it is distributed among the nodes, which validate the transaction and agree on the ledger's state via a distributed consensus mechanism. Distributed ledgers can be permissioned, that is, parties need permission to join the network. This ensures control over the number and identities of parties. In contrast, any party can join a permissionless ledger.

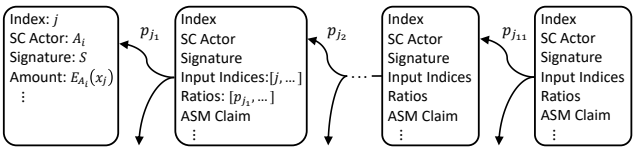


Fig. 3: Distributed Ledger Entries and Their Dependencies in the Cobalt Supply Chain (SC)

IV. Percentage Verification Protocol

A. Prerequisites

We use the index $1 \leq i \leq n$ for the supply chain actors A_i, e.g., miners, and the index $1 \leq j \leq m$ for these actors' private inputs, e.g., produced amounts. Each A_i is a certified member of the supply chain and has a key pair of a digital signature scheme. As described in Section II-A, we require a distributed ledger that reflects supply chain transactions.

Figure 3 illustrates a distributed ledger structure where each entry refers to its parent entries. These parent-child relations between supply chain transactions indicate which cobalt lots, commodities, or components a product is composed of. This allows for tracing the origin of cobalt lots by traversing the distributed ledger along the indices pointing to parent entries.

We distinguish between two different kinds of entries: ordinary entries and mining entries. Each entry contains a unique index j, a unique actor identifier A_i, and a signature proving that this entry was in fact written by A_i. Additional details such as timestamps, locations, assets types, etc. are possible but of no further interest for our protocol.

Mining entries, i.e., the initial entries of the supply chain, additionally contain an amount x_j of sourced cobalt ore in encrypted form. To enable light-weight mining entries, this amount can be published off-ledger, e.g., via distributed hash tables [23]. The unique identifier A_i implicitly tells whether this amount was sourced by an LSM or an ASM. In contrast to mining entries, all other entries additionally contain a list of indices pointing to parent entries that they were derived from as well as a list of parent-child ratios and a claim about the ASM percentage. The list of ratios contains one percentage value for each parent entry that the current entry points to. This allows for taking ratios between different lots of cobalt ore into consideration and therefore incorporates mixing of ingredients in the various supply chain steps, like in recipes. If we multiply all percentage values assigned to the parent-child references of the supply chain steps from an end product all the way back to a particular mined amount of cobalt ore, we learn what proportion this cobalt ore has in the end product.

We denote amounts of ASM cobalt ore by $x_{j_{ASM}}$ and amounts of LSM cobalt ore by $x_{j_{LSM}}$. Whether a mined amount x_j is an ASM amount, i.e., $x_{j_{ASM}}$, or an LSM amount,

1 :	$C \to RP$	θ
2 :	$C \to RP$	r_1, r_2
3 :	$DL \to RP$	$(..., E_{A_i}(x_j), ...)$
4 :	$DL \to RP$	$\forall j : (p_{j_1}, ..., p_{j_k}, ..., p_{j_d})$
5 :	RP	$(..., E_{DP}(x_j) = RE(E_{A_i}(x_j), rk_{A_i \to DP}), ...)$
6 :	RP	$\forall j : p_j = \prod_{k=1}^{d} p_{j_k}$
7 :	RP	$E_{DP}(S_{ASM}) = E_{DP}\left(\sum_{j=1}^{m} (x_{j_{ASM}} \cdot p_{j_{ASM}})\right) = \bigoplus_{j=1}^{m} (E_{DP}(x_{j_{ASM}}) \odot E_{DP}(p_{j_{ASM}}))$
8 :	RP	$E_{DP}(S_{Total}) = E_{DP}\left(\sum_{j=1}^{m} (x_j \cdot p_j)\right) = \bigoplus_{j=1}^{m} (E_{DP}(x_j) \odot E_{DP}(p_j))$
9 :	RP	$E_{DP}(S'_{ASM}) = E_{DP}(S_{ASM} \cdot r_3 + r_4) = (E_{DP}(S_{ASM}) \odot E_{DP}(r_3)) \oplus E_{DP}(r_4)$
10 :	RP	$E_{DP}(S'_{Total}) = E_{DP}(S_{Total} \cdot r_3 + r_4) = (E_{DP}(S_{Total}) \odot E_{DP}(r_3)) \oplus E_{DP}(r_4)$
11 :	$RP \to DP$	$E_{DP}(S''_{ASM}) = E_{DP}(S'_{ASM} + r_1) = E_{DP}(S'_{ASM}) \oplus E_{DP}(r_1)$
12 :	$RP \to DP$	$E_{DP}(S''_{Total}) = E_{DP}(S'_{Total} + r_2) = E_{DP}(S'_{Total}) \oplus E_{DP}(r_2)$
13 :	$DP \to C$	$S''_{ASM} = D_{DP}(E_{DP}(S''_{ASM}))$
14 :	$DP \to C$	$S''_{Total} = D_{DP}(E_{DP}(S''_{Total}))$
15 :	C	$\rho = \frac{S'_{ASM}}{S'_{Total}} = \frac{S''_{ASM} - r_1}{S''_{Total} - r_2}$

Protocol 1: ASM Percentage Verification Protocol

i.e., $x_{j_{LSM}}$, is determined implicitly by the miners identity A_i written in the mining entry (see Figure 3). We propose a protocol that computes the fraction of artisanal-mined cobalt ore as the target function

$$\rho = \frac{\sum_{j=1}^{m} x_{j_{ASM}}}{\sum_{j=1}^{m} x_{j_{ASM}} + \sum_{j=1}^{m} x_{j_{LSM}}} \tag{4}$$

(see Section II-A) in a privacy-preserving form via homomorphic encryption. That is, it divides the sum of incorporated ASM cobalt ore amounts by the total sum of incorporated cobalt ore amounts (ASM and LSM). The protocol involves three participants: the requesting consumer C who initiates the verification of a purchased product as well as a re-encryption party RP and a decryption party DP. The latter two carry out the majority of computations in the verification protocol but are no active contributors to the supply chain. We denote the distributed ledger by DL. The supply chain actors A_i do not actively participate in the verification protocol.

B. Setup

We require the following up-front key generations and key distributions. The decryption party DP has a key pair (pk_{DP}, sk_{DP}) of an asymmetric homomorphic encryption scheme (see Section III-A) with re-encryption capabilities (see Section III-B) and plaintext space $\mathcal{M}_{DP}$. Similarly, each A_i holds a key pair (pk_{A_i}, sk_{A_i}). The decryption party DP provides its pk_{DP} to RP and each A_i. Furthermore, each A_i generates a re-encryption key $rk_{A_i \to DP}$ and sends it to RP. This setup allows RP to transform the amounts encrypted under the individual pk_{A_i} into the same amounts encrypted under the common pk_{DP} by using the re-encryption keys $rk_{A_i \to DP}$. Given that, RP can then homomorphically evaluate the target function (see Equation (4)) on these ciphertexts without being able to decrypt them. Key distribution requires pairwise secure, i.e., secret and authentic, channels (see Section II-B).

The re-encryption party RP is necessary to allow encryption of amounts with individual keys while still enabling homomorphic evaluation of the target function on these amounts. The individual keys prevent unauthorized participants from reading actors' encrypted amounts. The additional decryption party DP provides the common key under which the homomorphic evaluation is performed and later decrypts the protocol result in a blinded form. Since DP is known at setup time, the re-encryption keys $rk_{A_i \to DP}$ only need to be generated by the A_i once for all verifications rather than individually for each verification. This not only causes constant costs of key generation and key distribution but also supersedes any involvement of the supply chain actors in the verification protocol. Hence, this setup with independent re-encryption and decryption parties ensures a high level of confidentiality and improves flexibility and performance.

C. Protocol Description

Prior to the verification protocol, in the j-th mining step, miner A_i encrypts the sourced amount x_j with pk_{A_i} and obtains the ciphertext $E_{A_i}(x_j)$. It publishes $E_{A_i}(x_j)$ in a new distributed ledger entry together with the index j, the ID A_i, and a signature for the entry. Later supply chain stages that use the cobalt ore mined in this step reference this j-th entry.

The verification process is depicted in Protocol 1. It consists of the following 15 protocol steps. We use "$\rightarrow$" to indicate communication between C, RP, and DP as well as interaction with the distributed ledger DL.

The protocol starts with the verifying consumer C sending to the re-encryption party RP the index θ of the distributed ledger entry that corresponds to the product that C wants to verify. Additionally, in step 2, C generates and sends two random numbers $r_1, r_2 \in \mathcal{M}_{DP}$ that will be used later for blinding the verification result that C receives.

Then, in step 3, RP traverses the distributed ledger DL starting from entry θ back to the mining entries that were used to create the product of interest. RP reads the encrypted amounts corresponding to these entries. We denote these encrypted amounts by $E_{A_i}(x_j)$ with $1 \leq j \leq m$ for m mining entries. During ledger traversal, RP also reads the parent-child percentage values p_{j_k} for each parent-child relation (step 4). In step 5, RP re-encrypts (see Section III-B) the encrypted amounts and obtains the amounts encrypted under DP's public key. Then, in step 6, RP multiplies all parent-child percentage values that correspond to the same mining entry, e.g., $p_{j_1}, ..., p_{j_k}, ..., p_{j_d}$ for supply chain depth d and mining entry j. As the cobalt supply chain typically has twelve stages [14], each mining entry has up to eleven child entries. Hence, we expect $d \leq 11$. We denote by p_j the overall percentage that a mined amount x_j accounts for in a product.

In steps 7 and 8, RP first homomorphically weights the mined amounts x_j by their percentages p_j (see Equation (3)). Then, RP homomorphically computes the encrypted weighted sum of the ASM cobalt ore amounts and the encrypted weighted total amount, i.e., ASM plus LSM amounts (see Equation (2)). We denote the resulting encrypted weighted sums by $E_{DP}(S_{ASM})$ and $E_{DP}(S_{Total})$, respectively.

Then, RP samples two random numbers r_3, r_4 s.t. $0 < r_4 \ll r_3 \in \mathcal{M}_{DP}$. These random numbers are used to additively and multiplicatively blind the encrypted sums S_{ASM} and S_{Total} in steps 9 and 10. They later ensure that C learns only the ratio rather than the exact sums of mined amounts. We denote the resulting encrypted blinded sums by $E_{DP}(S'_{ASM})$ and $E_{DP}(S'_{Total})$, respectively.

In steps 11 and 12, RP adds a second blinding layer to each sum by homomorphically adding the previously received r_1 and r_2, respectively. Then, RP sends the resulting encrypted, double-blinded sums $E_{DP}(S''_{ASM})$ and $E_{DP}(S''_{Total})$ to the decryption party DP. Additionally blinding the sums with r_1 and r_2 prevents DP from learning the ASM percentage.

DP decrypts the blinded sums and forwards the resulting blinded plaintexts to the consumer C in steps 13 and 14.

Eventually, in step 15, C removes the additive blinding values r_1 and r_2 by subtraction and divides the resulting S'_{ASM} by S'_{Total}. This quotient can then be compared to the claimed ASM percentage to determine whether the claim is correct or not. This concludes the verification protocol.

As long as $0 < r_4 \ll r_3$, i.e., r_3 is exponentially larger than r_4, multiplicatively blinding with r_3 preserves the ratio between S_{ASM} and S_{Total} since the r_3's in the quotient cancel out. The effect of r_4 renders factorization attacks to extract the sums practically infeasible. Thus, the consumer only learns the percentage of ASM cobalt ρ. This percentage is an approximate result due to the noise caused by additively blinding with r_4 (see steps 9 and 10). However, given $0 < r_4 \ll r_3$, the difference between ρ and the actual ASM percentage (see Equation (4)) can be expected to be negligibly small. We practically investigate this assumption in Section V-B.

D. Security Considerations

In our protocol, RP loads the amounts of cobalt ore in encrypted form (step 3). It homomorphically adds these ciphertexts to compute the encrypted sums without intermediate decryption (steps 7 and 8). These encrypted sums are blinded with the random r_1, r_2, r_3, r_4 (steps 9-12). Only after blinding, the sums are decrypted by DP. Therefore, RP processes the amounts and their sums in encrypted form and thus learns nothing about the amounts, sums, or the ASM percentage. Blinding the sums with r_1 and r_2 ensures that DP only learns random numbers and thus cannot derive the amounts, sums, or the ASM percentage. Blinding with r_3 and r_4 prevents C from learning the amounts or confidential sums. Hence, no one learns the single amounts or their sums and only the consumer learns the ASM percentage. Therefore, our protocol provides the desired confidentiality guarantees (see Section II-B).

V. Evaluation

We start evaluating our protocol by analyzing its asymptotic runtime. Then, we conduct an empirical analysis to verify the findings of the theoretical analysis and to demonstrate practicability of our protocol.

A. Theoretical Evaluation

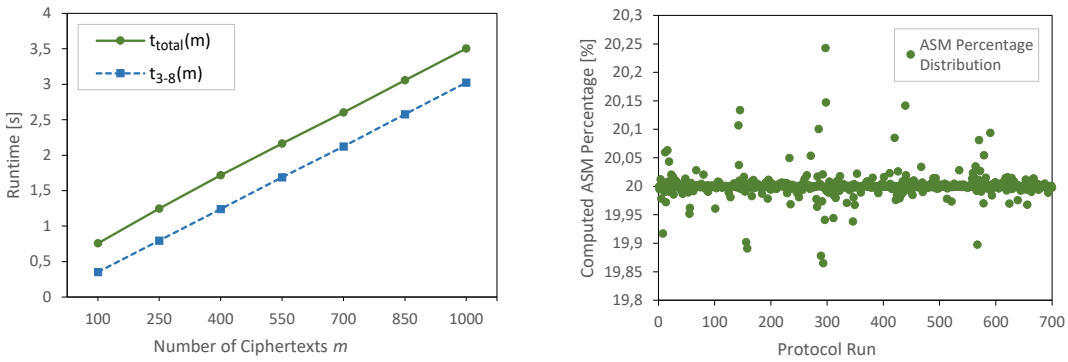
For the asymptotic runtime evaluation, we are going to apply standard Landau notation. Recall that the number of private inputs is denoted by m with index $1 \leq j \leq m$.

Proposition 1. *Protocol 1 has runtime in $\mathcal{O}(m)$.*

Proof. We are going to analyze the different types of computations applied in steps 1–15 of Protocol 1 individually. In particular, we are going to argue that each step has runtime in $\mathcal{O}(m)$, where we used $\mathcal{O}(1) \subset \mathcal{O}(m)$. Note that then the runtime of Protocol 1 is also in $\mathcal{O}(m)$. We recall the assumption of constant-size keys and entries, e.g., $|x_j|, |sk|, |pk| \in \mathcal{O}(1)$.

- The encryption and the decryption of s messages of constant size is done in $\mathcal{O}(s)$.
- The transmission of a message of size s is done in $\mathcal{O}(s)$.
- The generation of s random numbers is done in $\mathcal{O}(s)$.
- The homomorphic addition and the homomorphic multiplication of s terms is done in $\mathcal{O}(s)$.

Therefore, every step consisting of only these computations has runtime in $\mathcal{O}(m)$, which leaves steps 6 and 15. Due to the constant supply chain depth of twelve stages (see Section II-A), step 6 involves the multiplication of $\mathcal{O}(m)$ plaintext entries. Step 15 involves three plaintext operations. Thus, the plaintext computations in steps 6 and 15 have runtime in $\mathcal{O}(m)$, which concludes the proof. □



(a) Execution Time (b) Computed ASM Percentages

Fig. 4: Results of the Empirical Performance Analysis

B. Empirical Evaluation

We performed an empirical analysis to investigate whether the linear asymptotic runtime of our protocol ensures linear scalability. Moreover, we tested for practical applicability. Furthermore, we investigated the computed ASM percentages and compared them to the actual ASM percentages of the mining entries. This allows us to verify whether the effect of the additive noise r_4 in steps 9 and 10 is in fact negligible.

We built a prototype in Go and C++ and used the FHE scheme BFV [10] implemented in the PALISADE library,[1] which offers re-encryption capabilities. Where possible, we used more efficient ciphertext-plaintext operations, e.g., steps 7 and 8. We used a plaintext size of 59 bits, a security level of 128 bits, and set the remaining BFV scheme parameters as suggested in the Homomorphic Encryption Standard [3]. The 59-bit plaintexts can carry the annual global cobalt production (see Section I) in kilograms (28 bits) together with the blinding values r_1, r_2, r_3, r_4 (20-30 bits). As the distributed ledger, we used a permissionless Multichain,[2] which ensures compatibility with the Bitcoin ecosystem. We stored ciphertexts off-ledger and used caching to improve scalability of ledger traversal. We deployed the re-encryption party RP, decryption party DP, and consumer C in different locations across Europe. RP and DP where equipped with 32 3.1 GHz vCPUs while C ran on a machine with 2 2.2 GHz vCPUs and 2 GiB memory to match typical end consumer devices.

We measured the total verification runtime $t_{total}(m)$ as well as the time $t_{3\text{-}8}(m)$ spent on traversing the ledger and computing the weighted sums (steps 3-8), both for $m \in \{100, 250, 400, 550, 700, 850, 1000\}$ encrypted ASM and LSM amounts. We set the ASM percentage to 20 %. Figure 4a depicts the measured runtimes and shows that the total protocol runtime grows linearly in m. ASM percentage computation for 1000 mining entries took 3.50 s, while 100 ciphertexts were processed in 756 ms. The majority of that time is spent on traversing the ledger, loading the ciphertexts, and computing the weighted sums, represented by $t_{3\text{-}8}(m)$. These computations are highly parallelizable and therefore benefit from more computational power. The remaining time is spent on constant-time operations and communications.

Furthermore, we investigated the ASM percentage computed by our protocol. Figure 4b shows the computed percentages for all of our performed protocol runs. In most cases, the computed ASM percentage deviated less than 0.05 from the actual 20 % encrypted in the mined amounts. Hence, the effect of the noise used for protecting the summed amounts against factorization attacks (see Section IV-C) is in fact negligible.

Given the linear scalability and the fact that 1000 ciphertexts can be processed in just 3.50 s, we find our protocol to be suitable for practice. In the light of current efforts to increase efficiency of FHE schemes and their implementations, verification runtime can be assumed to decrease even further.

VI. Conclusion

An important aspect of ensuring ethical cobalt sourcing is to provide means to publicly verify the portion of cobalt ore mined in artisanal and small-scale mines (ASM). Performing this verification in a privacy-preserving manner by protecting processed amounts of cobalt can help supply chain actors to protect trade secrets and thus maintain competitive advantages.

We propose a cryptographic protocol that allows end consumers to verify ethical cobalt sourcing in terms of percentage of cobalt from ASMs used in their products. Its on-demand verification nature allows flexibility in the definition of ethical sourcing. The protocol augments existing distributed-ledger-based supply chain traceability systems by allowing verification by end consumers on product level. We employ a combination of homomorphic encryption and proxy re-encryption to guarantee confidentiality of cobalt ore amounts published on the distributed ledger. Our protocol runs in practically feasible time. We leave investigating the effect of other, potentially more efficient homomorphic encryption schemes and distributed ledgers to future work.

While ethical cobalt mining itself is a pressing problem, our proposed solution is not restricted to the verification of claimed ASM cobalt percentages. Instead, it can be applied to a variety of similar problems such as the amount of gold in an alloy or the percentage of fair-trade palm oil in groceries.

[1] https://gitlab.com/palisade/palisade-release
[2] https://www.multichain.com/

References

[1] Abbas Acar et al. "A survey on homomorphic encryption schemes: Theory and implementation". In: *ACM Computing Surveys (CSUR)* (2018).

[2] Tarun Kumar Agrawal. "Contribution to development of a secured traceability system for textile and clothing supply chain". PhD thesis. University of Borås, 2019.

[3] Martin Albrecht et al. *Homomorphic Encryption Security Standard*. Tech. rep. Toronto, Canada: HomomorphicEncryption.org, 2018.

[4] Amnesty International. *"This is what we die for": Human Rights Abuses in the Democratic Republic of the Congo Power the Global Trade in Cobalt*. 2016. URL: https://www.amnesty.org/download/Documents/AFR6231832016ENGLISH.PDF (visited on 07/13/2020).

[5] Matt Blaze, Gerrit Bleumer, and Martin Strauss. "Divertible protocols and atomic proxy cryptography". In: *Advances in Cryptology – EUROCRYPT'98*. 1998.

[6] Fabian Boemer et al. "NGraph-HE2: A High-Throughput Framework for Neural Network Inference on Encrypted Data". In: *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*. 2019.

[7] Miguel Pincheira Caro et al. "Blockchain-based traceability in Agri-Food supply chain management: A practical implementation". In: *2018 IoT Vertical and Topical Summit on Agriculture - Tuscany*. 2018.

[8] Center for Disease Control and Prevention. *Workplace Safety and Health Topics: Cobalt*. 2019. URL: https://www.cdc.gov/niosh/topics/cobalt/ (visited on 07/13/2020).

[9] Gaby G. Dagher et al. "Ancile: Privacy-Preserving Framework for Access Control and Interoperability of Electronic Health Records Using Blockchain Technology". In: *Sustainable Cities and Society* (2018).

[10] Junfeng Fan and Frederik Vercauteren. *Somewhat Practical Fully Homomorphic Encryption*. Cryptology ePrint Archive, Report 2012/144. https://eprint.iacr.org/2012/144. 2012.

[11] Craig Gentry. "A fully homomorphic encryption scheme". PhD thesis. Stanford University, 2009.

[12] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The Knowledge Complexity of Interactive Proof-Systems". In: *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*. 1985.

[13] Niclas Kannengiesser et al. "What Does Not Fit Can be Made to Fit! Trade-Offs in Distributed Ledger Technology Designs". In: *Proceedings of the 52nd Hawaii International Conference on System Sciences*. 2019.

[14] Barbara Lewis. "Blockchain to track Congo's cobalt from mine to mobile". In: *Reuters* (2018). https://www.reuters.com/article/us-mining-blockchain-cobalt/blockchain-to-track-congos-cobalt-from-mine-to-mobile-idUSKBN1FM0Y2. (Visited on 07/25/2020).

[15] Yehuda Lindell. *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*. 1st ed. Springer, 2017.

[16] Sidra Malik, Salil Kanhere, and Raja Jurdak. "ProductChain: Scalable Blockchain Framework to Support Provenance in Supply Chains". In: *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*. 2018.

[17] Edward McAllister. "Death toll rises at Glencore mine in Congo after collapse". In: *Reuters* (2019). https://www.reuters.com/article/us-congo-mining-glencore/death-toll-rises-to-43-at-glencore-mine-in-congo-after-collapse-more-expected-idUSKCN1TT1CT. (Visited on 07/13/2020).

[18] OECD. *OECD Due Diligence Guidance for Responsible Supply Chains of Minerals from Conflict-Affected and High-Risk Areas: Third Edition*. 2016. URL: https://www.oecd.org/daf/inv/mne/OECD-Due-Diligence-Guidance-Minerals-Edition3.pdf.

[19] Pascal Paillier. "Public-key Cryptosystems Based on Composite Degree Residuosity Classes". In: *Advances in Cryptology – EUROCRYPT'99*. 1999.

[20] Yuriy Polyakov et al. "Fast Proxy Re-Encryption for Publish/Subscribe Systems". In: *ACM Transactions on Privacy and Security* (2017).

[21] Ronald Rivest, Adi Shamir, and Leonard Adleman. "A Method for Obtaining Digital Signatures and Public-key Cryptosystems". In: *Communications of the ACM* (1978).

[22] Kim B. Shedd. *U.S. Geological Survey, Mineral Commodity Summaries: Cobalt*. 2020. URL: https://pubs.usgs.gov/periodicals/mcs2020/mcs2020-cobalt.pdf (visited on 07/13/2020).

[23] Ion Stoica et al. "Chord: A scalable peer-to-peer lookup service for internet applications". In: *ACM SIGCOMM Computer Communication Review* (2001).

[24] The World Bank. *The World Bank in DRC: Overview*. 2020. URL: https://www.worldbank.org/en/country/drc/overview (visited on 07/14/2020).

[25] Vivienne Walt. "Blood, Sweat, and Batteries". In: *Fortune Magazine* (2018). https://fortune.com/longform/blood-sweat-and-batteries/. (Visited on 07/14/2020).

[26] Martin Westerkamp, Friedhelm Victor, and Axel Küpper. "Blockchain-Based Supply Chain Traceability: Token Recipes Model Manufacturing Processes". In: *2018 IEEE International Conference on Blockchain (Blockchain)*. 2018.

[27] World Health Organization. *Concise International Chemical Assessment Document 69: Cobalt and Inorganic Cobalt Compounds*. 2006. URL: https://www.who.int/ipcs/publications/cicad/cicad69%20.pdf (visited on 07/13/2020).