L-013

Blockchain Empowered AI-Knowledge Graph Aggregation with Privacy-preserving Consensus

Guozhi Hao †　　Jun Wu †

1. Abstract

Since the high costs of creating knowledge graphs (KGs), trading and reusing KGs have profound implications for enabling AI services efficiently. However, there are some challenges for KGs trading: i) during KGs creation, poisoning attack causes consumers to get inaccurate results from KGs-based services. ii) in the process of training, dishonest server providers leak privacy, especially in sensitive areas such as medicine. To address the above issues, this paper proposes a novel architecture that securely constructs KGs with homomorphic consensus. Specifically, first, we devise KGs secure construction procedures on the proposed KGs consortium blockchain, which provides unmodifiable records to prevent unauthorized changes. Second, we propose a privacy-preserving KGs aggregation scheme that achieves consensus of encrypted KGs to prevent server providers from accessing privacy. Finally, experimental results of real KGs verify the market security and availability.

2. Introduction

Knowledge graphs (KGs) are becoming the essential technology in enabling advanced artificial intelligence (AI) services [1,2], from drug-drug interaction prediction to personalized recommendation, more and more researchers build KGs and mine the knowledge to achieve contributions in intelligent detection and analysis. The process of creating KGs is complicated and includes manual annotation, data cleaning, and other processes, which costs significant time and money. For domain KGs that solve similar problems, it is inefficient to create KGs repeatedly, so users need to trade and reuse KGs to promote the efficiency of research and industry.

KGs trading involves three parties, the creator, the service provider, and the consumer. The creator, usually some experienced expert, provides knowledge fact triples. Currently, KGs are usually constructed by a combination of manual and automatic methods [3]. Experts still need to manually develop high-quality knowledge relations for some domain knowledge graphs. Then, KGs service provider builds these fragmented fact triples into a knowledge graph and performs some necessary training and computation. To get more information, KGs usually require some training. Knowledge graph embedding (KGE) maps entities and relations into low-dimensional vectors

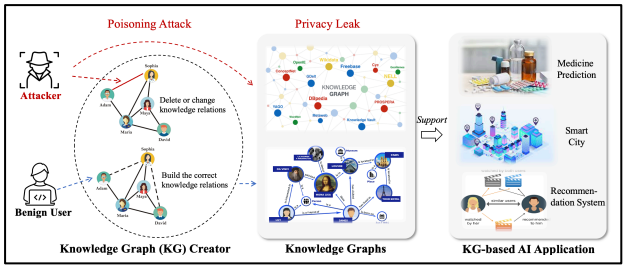

Fig. 1. Challenges of knowledge graph trading.

while capturing more KG semantic meanings, such as TransE [4], and TransH [5]. KGs service providers provide KGs-based services to consumers. However, there are some challenges for KGs trading as shown in Fig.1:

(1) Security risk: Researchers have proposed the poisoning attack method for KGs [6,7]. Since some knowledge relations still need to be manually marked, attackers can change the KGE results by adding or deleting specific knowledge relations. Then, KGs consumers get inaccurate KGs AI application results, such as wrong rating rankings of recommendation systems based on KGs.

(2) Privacy leakage: The KGs of some sensitive domains contain important privacy information, such as health [8,9]. When the KGs service provider conducts some computation tasks for these KGs, the creator and service consumer of KGs does not want the service provider to obtain private information.

Recently, blockchain has been widely used to build a decentralized privacy-preserving sharing system, which causes widespread concerns in both academia and industry. However, designing a secure architecture for knowledge graphs is still a novel challenge. Motivated by previous works, we exploit the blockchain technologies (consortium blockchain/smart contract), fully homomorphic encryption to build a secure KGs construction architecture, achieving KGs secure aggregation and privacy-preserving. The contributions of this paper are summarized as follows:

To implement secure KGs, we devise KGs secure decentralized construction on the proposed KGs consortium blockchain with smart contracts, which provides unmodifiable trading records to track and prevent unauthorized poisoning changes. To the best of our knowledge, it is the first time to propose a blockchain scheme for KGs to defend against poisoning attacks.

† Guozhi Hao and Jun Wu are with the Graduate School of Information, Production and Systems, Waseda University, Fukuoka 8080135, Japan.

To achieve privacy-preserving for KGs, we propose a privacy-preserving KGs aggregation scheme that achieves a homomorphic knowledge consensus of encrypted KGs to prevent server providers from accessing privacy, such as entity objects and relation weights. As we know, achieving consensus under encryption is the first time.

3. Problem Formulation

3.1 System Architecture

The architecture can implement the important processes during KG trading, which include privacy protection, secure and distributed management, construction and acquisition, aggregation, and application. The role of each entity is shown as follows.

KGs Creators (KGT): KGT creates the knowledge fact triples and relation weight from real-world big data and uploads them to KGP by the proposed KGs blockchain.

KGs Providers (KGP): KGP summarizes all knowledge facts and creates the global KG. As local servers, KGP nodes consist KGs blockchain and conduct relative computation.

KGs Consumers (KGC): KGC sends a request to KGP to trade KGs fact and application services such as some specific fact triples and link prediction results.

3.2 Threat Model

We consider a privacy-preserving architecture with anti-poisoning. In the KGs construction, analysis and trading, the potential threats are shown as follows:

Poisoning Attacks: For traditional KGs databases, the permission to modify KG is open to the public or experts, malicious KGT clients upload elaborately malicious fact triples and attempt to change the result of KGE. Besides, for domain-specific KGs, knowledge facts are usually written by domain experts who have higher authority and can change the KG as they wish. The goal of malicious KGT clients is to influence the performance of the KGE-based AI application without being detected.

Privacy Leakage: As the provider of KG's computing and services, the KGP are honest but curious, which means that they honestly execute the established protocols but may be curious to deduce some sensitive information. Some KGP nodes leak the obtained KGs data, especially in some privacy-sensitive areas, such as healthcare.

4. Design of Proposed Scheme

4.1 Technical Overview

As shown in Fig. 2, our proposed scheme is based on blockchain technology, where knowledge aggregation, management, and consensus are all performed in the proposed distributed architecture. To protect the privacy information, the KGs need to be encrypted when they are sharing and trading in KGB. KGT is responsible for creating knowledge relations, encrypting the knowledge triples f and weights w using FHE, and then publishing the fact

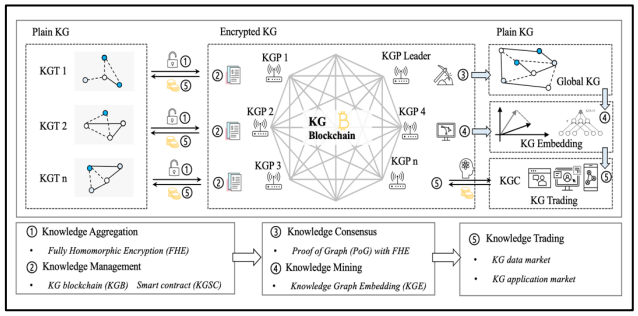

Fig. 2. The overview of proposed KGs construction scheme for KGs trading.

transactions through the smart contract KGSC. KGP nodes form KGB to receive encrypted knowledge relations cf and construct KGs. After the proposed FHE-based consensus PoG, the block leader publishes the global KGs of this round, KGP nodes conduct KGE computation and provide application services according to global KG $G = \{E, R, F\}$.

4.2 Secure Knowledge Graph Aggregation

Our proposed scheme uses fully homomorphic encryption and smart contract to realize secure knowledge aggregation and management for KGs. In the following, we describe the details of the processes.

As shown in Fig. 3, KGT uses IDs e_i and r_i to represent real-world entities and relations, such as 441 for 20th

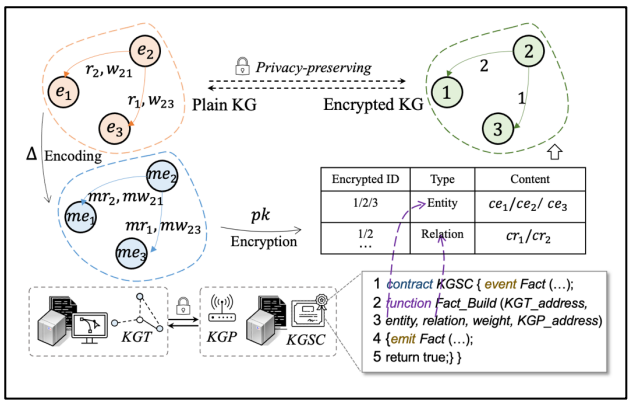

Fig. 3. Proposed secure KGs aggregation.

Century Studios in KGs dataset FB15K237. These IDs form the plain KGs fact f with the risk of privacy leakage. So, in our proposal, KGT encodes the plain fact $f = (e_i, r_k, e_j, w_{ij})$ to $mf = (me_i, mr_k, me_j, mw_{ij})$ with parameter Δ and then encrypts it by public key pk to $cf = (ce_i, cr_k, ce_j, cw_{ij})$ by CKKS scheme. Then, KGT sends these encrypted facts cf to the nearest KGP node. KGP node numbers the received cipher facts, denotes entity as IDe, and denotes relations as IDr. Each entity and relation correspond to a unique ID, and all KGP nodes in the KGB jointly maintain mapping tables Ei and Ri to record links between cipher KGs IDs. KGP publishes the encrypted facts into the blockchain via the smart contract KGSC. Figure 3 shows the main structure of KGSC. Basically, the ID of the encrypted fact with the addresses of KGT and

KGP are used as input in the smart contract for later consensus based on the graph content. Besides, KGSC emits an event to make this construction process public and allow KGB participants to review it.

For KGPs, the content of the KGs are IDs of ciphertext, as computation service providers, KGPs do not know the real information corresponding to the graph. Importantly, the existing graph structure information can already support KGs analysis and management by KGPs. For example, for the KGs-based recommendation system, KGP get the top-rated entities and relations IDs, but does not know the corresponding real information. After KGP sends these cipher texts to KGC, the original text corresponding to the real-world information can be obtained after decryption. After KGs construction process is published, KGP nodes are also unable to delete or change the published knowledge graphs, resisting poisoning attacks.

4.3 Knowledge Homomorphic Consensus for KGs

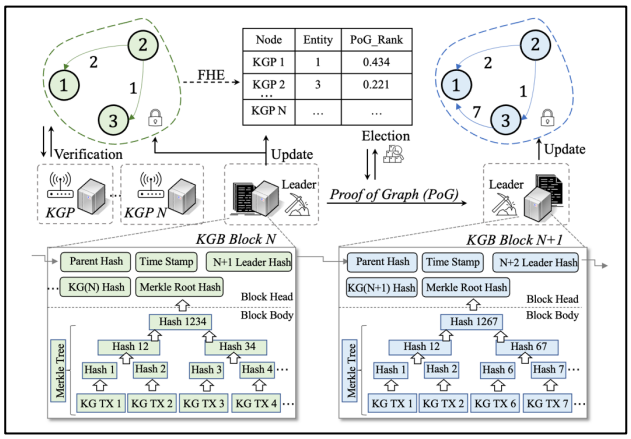

Fig. 4. Homomorphic consensus for KGs.

After receiving the new facts in the current block update cycle, the KGP nodes in the KGB broadcast these new transactions to each other, and each node modifies locally to get the global graph $G = \{E, R, F\}$ for the current block. We develop a traditional Pagerank algorithm [10] to evaluate the importance of the nodes in the graph. Specifically, we innovatively introduce weight values and design the algorithm in the form of a fully homomorphic encryption computation. The node's score is assigned to each KGP node through the proposed allocation mechanism, and the KGP node with the highest score is responsible for publishing the global KG for the round and updating the blocks. We describe the details of the processes as follows.

As shown in fig. 4, the updated graph in this round performs a fully homomorphic encryption computation to obtain the importance score ER of each entity. For a KG, this computation is updated iteratively, and the iterative process of scoring the entity e_i without encryption for trun T is shown in equation:

$$ER_{e_i}(T) = s\sum_{j=1}^{N_i}\frac{w_{ji}ER_{e_j}(T-1)}{k_j} + (1-s)\frac{1}{N} \quad (1)$$

For entity e_i, suppose there are N entities in KG $G = \{E, R, F\}$, iterating through all relations pointing to e_i, the importance of e_i is derived from the other entities e_j pointing to it. The more important these entities e_j are, the higher the weight w_{ji} of the relation, and the higher the importance ER_{e_i}. s is a random wandering coefficient to avoid local convergence of the algorithm. As shown in equation $ER_{KGP_k} = \sum_{i=1}^{N_{KGP_k}} ER_{e_i}$, for KGP node KGP_k, consensus ranking scores are derived from attributed entities.

In order to protect privacy, KGP only get the encrypted KG, we design the homomorphic computation of the above equation based on the fully homomorphic encryption scheme CKKS, as shown in equation:

$$ER(T) = (r_{c1} \otimes r_{p1}) \oplus (r_{c2} \otimes r_{p2}) \quad (2)$$

KGP can perform node importance computation and complete consensus by ciphertext without knowing KGs content. The entities importance are represented by the vector $\boldsymbol{ER}(T)$, which is obtained by homomorphic addition of the results of two homomorphic multiplications. To avoid homomorphic computational errors that prevent decryption to get the correct result, a homomorphic multiplication is performed by a ciphertext term $\mathbf{r_c}$ with a plaintext term $\mathbf{r_p}$, and the results of the two sets of multiplications are then homomorphically added.

Specifically, $r_{c1} = [r_{c1}^1, r_{c1}^2, \dots, r_{c1}^n]$ represents ciphertext of relationship weights, $r_{p1} = [r_{p1}^1, r_{p1}^2, \dots, r_{p1}^n]$ updates every round as shown in equation:

$$r_{p1}(T) = s \cdot r_{p1}(T-1)/k_e \quad (3)$$

which represents previous round entity score. $r_{c2} = [r_{c2}^1, r_{c2}^2, \dots, r_{c2}^n]$ stores the location of the entity as the relation head, and $r_{p2} = [r_{p2}^1, r_{p2}^2, \dots, r_{p2}^n]$ stores random wandering items as shown in equation:

$$r_{p2} = r_{c2} \cdot (1-s)/N \quad (4)$$

Based on the updated KG from the previous block, PoG select the KGP with the highest score as the leader for this round of block. As shown in fig. 4, the leader publishes new global KG based on KG transactions and conduct consensus computation. The blockhead contains the hash algorithm results of KG to prevent KG and consensus results from being tampered, such as leader address and KG transaction Merkle tree. All KGP nodes can check the transaction records to verify whether the graph content and consensus are legal.

5. Security Analysis and Performance Simulation

5.1 Setting and Dataset

We use Microsoft's Simple Encrypted Arithmetic Libary (SEAL) [11] to deploy CKKS [12] and implement our proposed blockchain homomorphic consensus scheme PoG. We use real-world KGs dataset FB15K237 to conduct our proposed aggregation scheme, there are 237 relations, 14541 entities, and 272115 triples in train datasets. The experiment is deployed on a server with Ubuntu 20.04LTS,

CPU Xeon Gold 6326 2.9GHz, 256G DDR4 RAM, 8TB SSD and GPU NVIDIA RTX A5500.

5.2 Blockchain-based KGs Encryption and Analysis of Anti-poisoning

To show the process of KGs secure aggregation. We selected a section of the FB15K237 about movies. As shown in Table 1, each real concept has a Plain KG ID and our scheme create new random encrypted KG ID for them to protect privacy. KGP only get encrypted KG ID and do not know the real content of them. We can find in Table 1 that each real-world fact contains four values, including head entity, relation, tail entity, and weight. The movie Aliens won Academy Award for Best Sound and this real-world fact can be presented by (371, 7, 4411, 1). Then, our proposed scheme creates encrypted facts for each real-world fact. Entities and relations use random encrypted KG

Real world entity/realition/fact	Plain KG ID	Encrypted KG ID
20th Century Studios	441	1
Star Wars: Episode IV – A New Hope	1195	2
Monty Python and the Holy...	5044	3
Academy Award for BS	371	4
Aliens	4411	5
Academy Award for Best Writing, Original Screenplay	1654	6
film_distributor	32	1
award	7	2
Aliens won Academy Award for Best Sound	(371, 7, 4411, 1)	(4, 2, 5, cw_{45})
Star Wars: Episode IV won Academy Award for BWOS	(1654, 7, 1195, 1)	(6, 2, 2, cw_{62})
...	...	...

Table. 1. KGs fact mapping.

ID and the weight are encrypted by CKKS. KGPs use encrypted KG ID and encrypted weight value to conduct computation.

We evaluate the anti-poisoning performance of the proposed scheme in Fig. 5. The poisoning attack methods on the KGs are deleting and modification, we performed both attacks in the FB15K237 with different poisoning rates

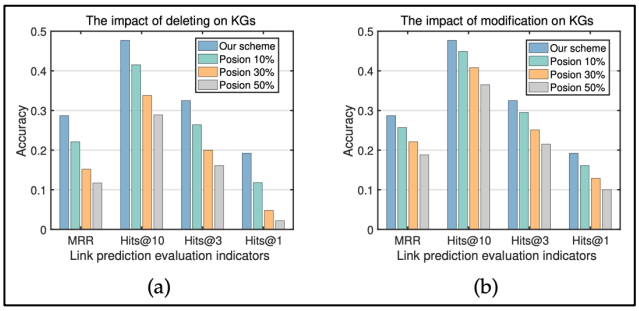

Figure. 5. Performance of anti-poisoning

and then used the dataset for a link prediction task. As shown in Fig. 5, the poisoning attack reduces the accuracy of link prediction, and the more graphical facts of the attack, the lower the accuracy. In our solution, the blockchain records all the operations of creating the KGs, so that the consensus and published KGs cannot be changed. The results show that our scheme avoid the decrease in the accuracy of link prediction caused by poisoning.

6. Conclusion

In this paper, we proposed a privacy-preserving knowledge graph construction scheme for KGs trading with anti-poisoning capability to resist poisoning attacks, and protect the privacy of KG. We use a real-world KG dataset to conduct the experiment. For evaluation, we show the encryption of a movie KG in our proposed blockchain, and we demonstrate the performance of the proposed scheme under two poisoning attacks. In the future, we will explore low consumption, high transaction speed blockchain knowledge sharing system, and privacy protection issues for consensus and smart contracts.

Acknowledgment

This work was supported in part by the JSPS KAKENHI under Grants 23K11072, in part by the National Natural Science Foundation of China under Grants U21B2019 and 61972255.

Reference

[1] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, “A survey on knowledge graphs: Representation, acquisition and applications,” IEEE Transactions on Neural Networks, 2021.

[2] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, “A review of relational machine learning for knowledge graphs,” arXiv: Machine Learning, 2015.

[3] G. Weikum and M. Theobald, “From information to knowledge: harvesting entities and relationships from web sources,” symposium on principles of database systems, 2010.

[4] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” neural information processing systems, 2013.

[5] Z. Wang, J. Zhang, J. Feng, and Z. Chen, “Knowledge graph embedding by translating on hyperplanes,” national conference on artificial intelligence, 2014.

[6] H. Zhang, T. Zheng, J. Gao, C. Miao, L. Su, Y. Li, and K. Ren, “Data poisoning attack against knowledge graph embedding,” in Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019, pp. 4853–4859.

[7] P. Banerjee, L. Chu, Y. Zhang, L. V. Lakshmanan, and L. Wang, “Stealthy targeted data poisoning attack on knowledge graphs,” in 2021 IEEE 37th International Conference on Data Engineering (ICDE), 2021, pp. 2069–2074.

[8] T. Ma, X. Lin, B. Song, P. S. Yu, and X. Zeng, “Kg-mtl: Knowledge graph enhanced multi-task learning for molecular interaction,” IEEE Transactions on Knowledge and Data Engineering, pp. 1–12, 2022.

[9] A. Fernández-Torras, M. Duran-Frigola, M. Bertoni, M. Locatelli, and P. Aloy, “Integrating and formatting biomedical data as pre-calculated knowledge graph embeddings in the bioteque,” Nature communications, vol. 13, no. 1, pp. 1–18, 2022.

[10] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web.” Stanford InfoLab, Tech. Rep., 1999.

[11] “Microsoft SEAL (release 4.1),” https://github.com/Microsoft/SEAL, Jan. 2023, microsoft Research, Redmond, WA.

[12] J. H. Cheon, A. Kim, M. Kim, and Y. Song, “Homomorphic encryption for arithmetic of approximate numbers,” in International conference on the theory and application of cryptology and information security. Springer, 2017, pp. 409–437.