A Comprehensive Solution for Integrity Management and Access Control of Document Storage in Blockchain Network

H M Shahriar[1] and Muhammad Nur Yanhaona[2]

***Abstract*—**

I. INTRODUCTION

A. *Context*

B. *Problem*

C. *Current landscape of solution alternatives*

D. *Solution overview*

E. *Paper Organization*

II. SCOPE OF DOCUMENT STORAGE INTEGRATION PROBLEM

A. *Integrity*

We have used SHA-256 of document content hash to ensure the integrity of the document. Document hash will be stored in blockchain. External module will calculate the hash from downloaded document and verify with the stored hash to ensure the integrity of the document.

B. *Access Control*

The Access controller works as an interface between the external document storage and the user during document upload and download. It stores the external storage information of uploaded document where it actually exists. It performs cryptographic operations to provide the confidentiality of a document from other users of the system. The Access controller also provides a permission control mechanism during view and download of a document by a user. Only intended user for a document in the system can download and view the document.

C. *Payment Integration*

We introduce a generic fee calculation system which will calculate the upload fee of a document according to the size of the document. The fee calculation system also calculates the download payment for a user during the download operation.

*This work was supported by Kona Software Lab Ltd

[1]H M Shahriar, Sr Engineer, Kona Software Lab Ltd, Dhaka, Bangladesh `h.m.shahriar@konasl.com`

[2] Muhammad Nur Yanhaona, Sr Engineer, Kona Software Lab Ltd, Dhaka, Bangladesh `nur.yanhaona@konasl.com`

III. SYSTEM ARCHITECTURE

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

A. *components*

User, Blockchain, Access Controller, External storage

B. *Roles*

- User:
- Access Controller:
- Blockchain
- External Storage

C. *Interaction*

The user interacts with the Access controller and the blockchain directly using DAPP to upload and download a document. The user provides blockchain information to the access controller to validate and verify the document information. The Access controller verifies the user information and performs blockchain transaction according to the provided information. Access controller communicates with external storage and generates session during document download.

IV. DOCUMENT UPLOAD PROTOCOL

The document upload protocol is used to upload a document to the system. Upload protocol is composed of three main phases. The User collects a payment token from the *Access Controller* and deposits upload fee to the blockchain for the *Access Controller*. The *Access Controller* verifies the payment and uploads the document in external storage. The User verifies the upload process and initiates fee transfer operation for the *Access Controller*.

Full details of each phase is described below:

A. Token Generation

The User generates a unique document key D_k with document information (document name D_n, document uploader blockchain address DU_{addr} and document container contract address DC_{addr}) to identify each document.

$$D_k = hash(D_n, DU_{addr}, DC_{addr}) \tag{1}$$

The User requests the *Access Controller* to generate a token for uploading a document with document size D_s, document hash D_h, signature of document hash DH_{sig}, DU_{addr} and D_k. The *Access Controller* validates the input data from user and verifies the signature using an already deployed smart contract (responsible to perform sign verification) from the blockchain node it is connected with. The *Access controller* generates a secret key K_1 and encrypts its blockchain address AC_{addr} with the generated key and produces an payment token M.

$$M = Enc(AC_{addr}, K_1) \tag{2}$$

The Access controller calculates the document upload fee according to the size of the document, stores the secret key K_1 for further use in next upload phase and returns the generated token M and calculated $fee\ amount$ to the User.

B. Request Upload

In this phase, the user generates a secret key K_2, encrypts the payment token M with the secret key and produces an upload token N.

$$N = Enc(M, K_2) \tag{3}$$

The user performs a transaction to the blockchain with token M and N and the document information (D_n, D_h and other document metadata). During this operation, the calculated upload fee to upload the document is also deposited to the blockchain. After successful mining of the transaction, the user sends the document upload request to the *Access Controller* with actual document and its metadata.

The *Access controller* collects the payment token M from the blockchain and verifies that if it contains the access controller's address by performing a decryption on token M with key K_1 and checks that if the result holds the following condition.

$$AC_{addr} = Dec(M, K_1) \tag{4}$$

The Access controller also verifies the document size and deposited fee from the blockchain. If all verifications succeed, it uploads the actual document to an external storage and stores the external storage information. Now the user can verify the document upload status by issuing a request to the *Access Controller*.

C. Finalize Upload

After a satisfactory verification of the outcome of upload, the user issues an unlock payment transaction with the user secret key K_2 to the blockchain. The blockchain verifies the key by decrypting N with provided K_2 and matches with stored payment token M. If the verification succeed, the smart contract unlocks the payment to forward the amount to the receiver address encrypted as payment token M.

$$M = Dec(N, K_2) \tag{5}$$

The user requests the *Access Controller* to finalize the document upload procedure with document information, signature and secret key K_2. The *Access Controller* verifies the signature of the user and checks the payment unlock status from the blockchain. The *Access Controller* issues a transaction to collect the upload payment with its secret key K_1. The smart contract decrypts payment token M with K_1 and matches the result with the caller address. If the address matches with decrypted result, smart contract transfers the payment to the caller (Access Controller) address.

$$AC_{addr} = Dec(M, K_1) \tag{6}$$

V. Document Download Protocol

The Document Download Protocol is used to download a document from the blockchain. The document download protocol is a combination of four main phases. The User collects a payment token from the *Access Controller* and deposits download fee to the blockchain for the *Access Controller*. The *Access Controller* collects a download session from the external storage, perform some cryptographic operations and returns an encrypted session to the user. The User checks the transaction from the blockchain, unlocks the download payment and initiates a fee transfer request to the *Access Controller*. The user will collect the encrypted session encryption key from the blockchain and regenerates the download session by performing cryptographic operations on it, and downloads the document from the external storage directly by using the session.

The working procedure of each phase can be subdivided into different steps. The required operations to perform different steps is described below.

A. Download Token Generation

The user requests to the *Access Controller* to generate a payment token for downloading a document with document key D_k, document hash D_h, user blockchain address U_{addr} and signature of document hash DH_{sig}. The *Access Controller* receives and validates the input data and verifies the signature DH_{sig} using D_h as message. The *Access Controller* generates a payment secret key K_a and encrypts its blockchain address AC_{addr} with the secret key and produces an payment token T_p.

$$T_p = Enc(AC_{addr}, K_a) \tag{7}$$

The *Access Controller* also calculates the download fee to download the document, stores the secret key K_a and returns the payment token T_p with calculated download fee amount to the user.

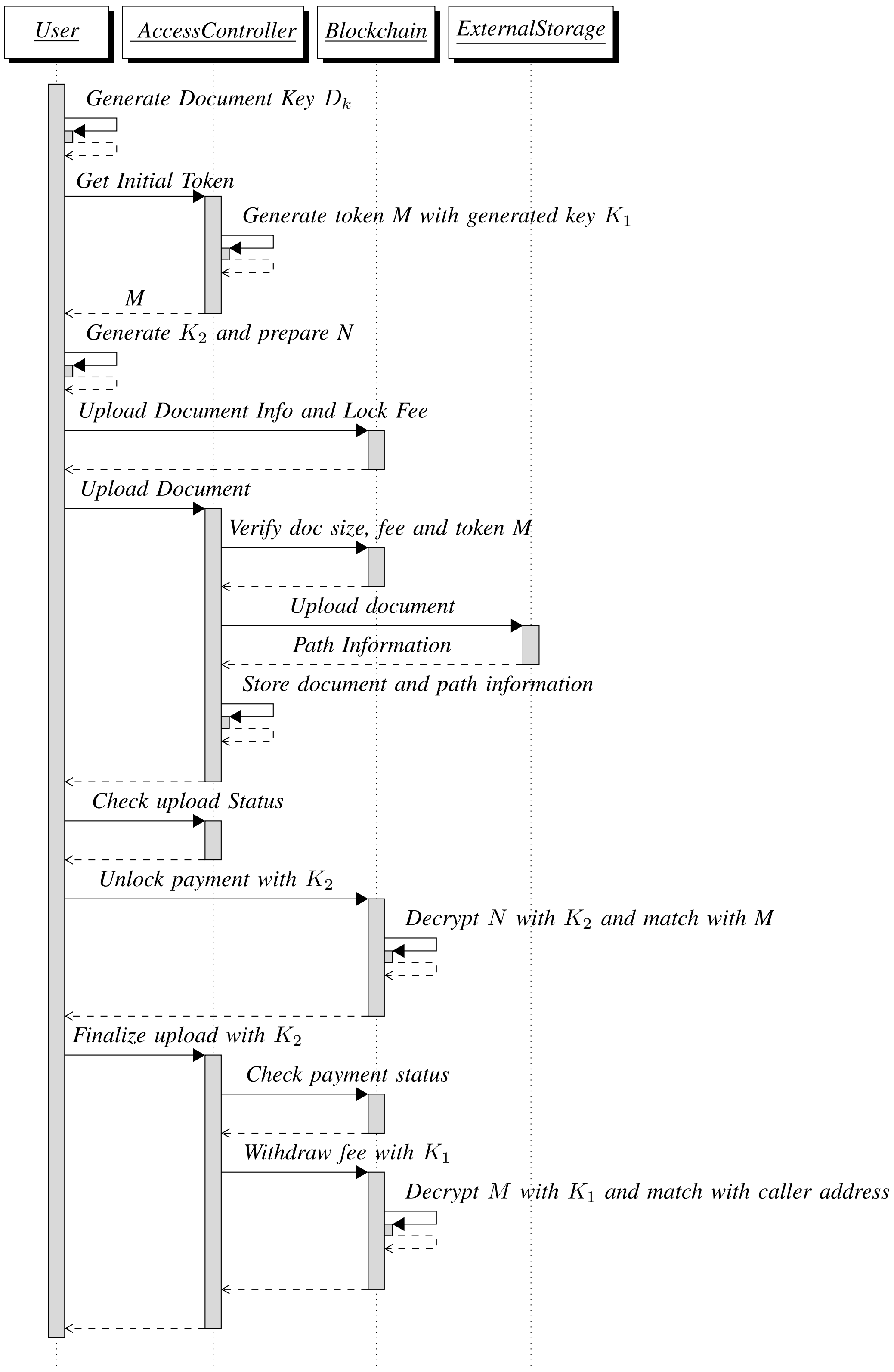
User
AccessController
Blockchain
ExternalStorage
Generate Document Key D_k
Get Initial Token
Generate token M with generated key K_1
M
Generate K_2 and prepare N
Upload Document Info and Lock Fee
Upload Document
Verify doc size, fee and token M
Upload document
Path Information
Store document and path information
Check upload Status
Unlock payment with K_2
Decrypt N with K_2 and match with M
Finalize upload with K_2
Check payment status
Withdraw fee with K_1
Decrypt M with K_1 and match with caller address

Fig. 1. *Sequence diagram of upload protocol*

B. *Prepare Download Session*

After receiving the payment token, the user generates two secret keys K_c and K_u. K_c is the common secret, shared only with the *Access Controller*. The user generates an unlock token T_u by performing encryption operation on T_p with the key K_u.

$$T_u = Enc(T_p, K_u) \tag{8}$$

The user issues a download payment transaction to the *blockchain* with two tokens T_p and T_u, document information and deposits the payment amount. After successful mining of download payment transaction, the user issues a prepare download session request to the *Access Controller*. During this request the user sends document container contract address DC_{addr}, user address U_{addr}, document key D_k, document hash D_h, signed document key DK_{sig} and the K_c to the *Access Controller*. The *Access Controller* validates the input data, and also verifies the sign data and payment status from the *blockchain*. In return of successful verification of payment status, the *blockchain* returns the payment token T_p to the *Access Controller*. The *Access Controller* then verifies the T_p by performing decryption operation on it with K_a and matches the result with its own blockchain address.

$$AC_{addr} = Dec(T_p, K_a) \tag{9}$$

After a successful address verification the *Access Controller* retrieves the external storage information for the document and requests the *External Storage* for a session S to download the document. The *Access Controller* generates a secret key K_s, encrypts S with the secret key and produces an encrypted session ENC_s. It also encrypts the document key with the common secret K_c and produces an encrypted document key ENC_{dk}.

$$ENC_s = Enc(S, K_s) \tag{10}$$

$$ENC_{dk} = Enc(D_k, K_c) \tag{11}$$

After performing encryption operations the *Access Controller* prepare a hash of the encrypted session and issues a transaction to the *Blockchain* with these hash value and ENC_{dk}. The *Access Controller* stores all the input data and returns the transaction hash and ENC_s to the user.

C. *Get Encrypted Session*

The user issues a transaction to the blockchain to unlock the payment with K_u, D_k and U_{addr}. The *Blockchain* unlocks the fee amount performing decryption on the unlock token T_u with the secret key K_u provided by the user and matches the result with the payment token T_p. The *Blockchain/* unlocks the payment amount to withdraw, if the verification of token satisfies the following condition.

$$T_p = Dec(T_u, K_u) \tag{12}$$

The user issues a request to the *Access Controller* to get the download session with D_k, DK_{sig} and U_{addr}. The *Access Controller* verifies the signature, checks payment unlock status from the *Blockchain*. After a successful verification, the *Access Controller* encrypts the session secret K_s with common secret K_c and produces an encrypted session encryption key ENC_{ks}.

$$ENC_{ks} = Enc(K_s, K_c) \tag{13}$$

The *Access Controller* issues a transaction to the *Blockchain* to withdraw the payment with the *Access Controller* secret K_a and ENC_{ks}. Blockchain verify the secret K_a by decrypting the payment token T_p with K_a and match the result with the caller address. If the decrypted result matches the caller address, blockchain transfers the download payment fee to the caller address. It also stores the ENC_{ks} with the download document information. The *Access Controller* returns the transaction hash to the user. After successful mining of the transaction, the User collects the encrypted session encryption key ENC_{ks} from the blockchain.

D. *Download Document*

The User first decrypts the encrypted session encryption key ENC_{ks} with the common secret K_c and retrieves the session secret key K_s, then decrypts the encrypted session ENC_s with K_s to retrieve the original session S.

$$K_s = Dec(Enc_{ks}, K_c) \tag{14}$$

$$S = Dec(Enc_s, K_s) \tag{15}$$

Now the User can get the document directly from the External Storage using the session S.

VI. System Analysis

In this section, we analyze the security measures and privacy policy of our proposed protocol systems.

A. *Access Controller Perspective*

1) Accountability: Upload and download protocol involves interactions among user, access controller, blockchain and external storage, where blockchain ensures accountability as any user can retrieve any transaction stored in it. In the proposed protocols, all the payment transactions and document metadata is stored in blockchain, so anyone can verify the operations performed in this protocol, as the operations are transparent and available.

2) Protection from external storage system failure: Payment is done only after all interactions with the external storage are complete

3) Guaranteed payment: In both upload and download protocol, payment token is generated by the *Access Controller*, performing an encryption operation on its blockchain address which confirms the receiver of the payment is *Access Controller*. *Access Controller* always performs payment verification before performing upload a document or retrieve download session from the *External Storage* which ensures the payment for the *Access Controller*. In our proposed protocols transfer the payment fee only possible when caller pass a secret key (K_1 for upload and K_a for download protocol) that can decrypt the payment token and the result matches the caller address. This process guaranteed that only the *Access Controller* can receive the payment fee.

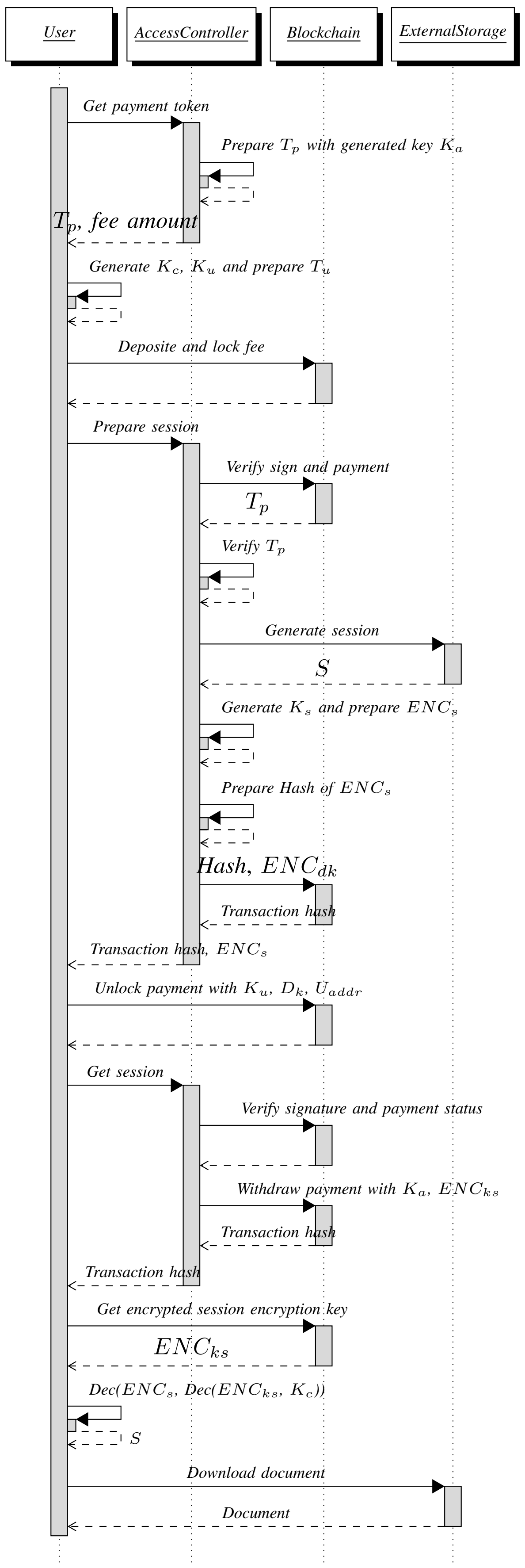
User
AccessController
Blockchain
ExternalStorage
Get payment token
Prepare T_p with generated key K_a
T_p, fee amount
Generate K_c, K_u and prepare T_u
Deposite and lock fee
Prepare session
Verify sign and payment
T_p
Verify T_p
Generate session
S
Generate K_s and prepare ENC_s
Prepare Hash of ENC_s
Hash, ENC_{dk}
Transaction hash
Transaction hash, ENC_s
Unlock payment with K_u, D_k, U_{addr}
Get session
Verify signature and payment status
Withdraw payment with K_a, ENC_{ks}
Transaction hash
Transaction hash
Get encrypted session encryption key
ENC_{ks}
Dec(ENC_s, Dec(ENC_{ks}, K_c))
S
Download document
Document

Fig. 2. *Sequence diagram of download protocol*

4) Minimization of state management: Most of the state information remains in the blockchain and failure cases and chain reorganization can be dealt with efficient

5) Protection from mining network: Operations in blockchain is public and the final inclusion of transaction is performed by the miner, who validates the transaction and combines transactions to prepare and add a block in the blockchain. So miner also have the knowledge of the secret key of *Access Control* before adding the transaction in the blockchain. Miner can also submit a similar transaction with the secret key before processing the actual transaction to receive the payment. But the knowledge of secret key is not beneficial for the miner as the encrypted payment token holds the *Access Control* address. It protects the system from miners to issue similar fraudulent payment transaction.

B. User Perspective

1) Security protection: In our proposed protocols user has the capability to unlock the payment. Without unlocking the payment Access Controller can't withdraw the payment for upload/download from the blockchain. In this system user can verify his/her requested service before unlock the payment in blockchain. It also ensures that user is able to get refund his/her payment if he/she does not get the proper service. The operations performed during document upload and download include cryptographic operations which ensures none else the user can interrupt or sniff on the storage session intended for the user.

2) Payment enforcement: User must pay before he/she gained access to the external storage and his/her access can be properly restrained or monitored

3) Protection from the mining network maliciousness: External storage related information cannot be derived from the blockchain data

VII. Access Configuration & Control

VIII. Implementation Concern

Time constrain for download session: If mining takes more time than the validity of download session, user can never get the document downloaded. So the system need to ensure the mining fast or update the external storage session validation time.

A. Event Processing

B. Maintain Database

C. Timing and Synchronization

IX. Related Work

A. Blockchain Based Storage

B. External Storage Integration

X. CONCLUSIONS

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

APPENDIX

Appendixes should appear before the acknowledgment.

ACKNOWLEDGMENT

The preferred spelling of the word acknowledgment in America is without an e after the g. Avoid the stilted expression, One of us (R. B. G.) thanks . . . Instead, try R. B. G. thanks. Put sponsor acknowledgments in the unnumbered footnote on the first page.

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

REFERENCES

[1] G. O. Young, Synthetic structure of industrial plastics (Book style with paper title and editor), in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 1564.

[2] W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123135.

[3] H. Poor, An Introduction to Signal Detection and Estimation. New York: Springer-Verlag, 1985, ch. 4.