МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ОДЕСЬКА НАЦІОНАЛЬНА АКАДЕМІЯ ХАРЧОВИХ ТЕХНОЛОГІЙ

ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ І ТЕХНОЛОГІЙ «ІНДУСТРІЯ 4.0» ІМ. П.Н. ПЛАТОНОВА

XII МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ І АВТОМАТИЗАЦІЯ – 2019

INFORMATION TECHNOLOGIES AND AUTOMATION– 2019

Збірник доповідей

Частина II

Одеса,
17-18 жовтня 2019

Секція 2

Наукові напрямки:

Сучасні методи і алгоритми управління об'єктами хіміко-технологічного типу

Автоматичні і автоматизовані системи управління технологічними процесами харчової та зернопереробної промисловості

Автоматизоване управління бізнес-процесами: концепції, методи, алгоритми, системи

Штучний інтелект і автоматизація робототехнічних систем

Нове в розвитку інформаційно-керуючих технологій: технічна база, програмне забезпечення, мережі.

Список
скорочень організацій, представники яких взяли участь у конференції

Таблиця 1

Скорочення	Повна назва організації	Місто	Країна
BNTU	Belarusian National Technical University	Minsk	Belarus
CAFU	CRIAME of Armed Forces of Ukraine	Kyiv	Ukraine
DMTSAU	Dmutro Motornyi Tavria State Agrotechnological University	Melitopol	Україна
DNU	Vasyl' Stus Donetsk National University	Вінниця	Україна
EKSTU	East Kazakhstan State Technical University D. Serikbayev	Ust-Kamenogorsk	Kazakhstan
IAEl SB RAS	Institute of Automation and Electrometry of the Siberian Branch of the Russian Academy of Sciences	Novosibirsk	Russia
IRTC IT&S NAS AND MES	International Research and Training Center for Information Technologies and Systems of the National Academy of Sciences (NAS) of Ukraine and Ministry of Education and Science (MES) of Ukraine	Kyiv	Ukraine
KGES	Kharkiv general education school	Kharkov	Україна
LPNUU	Lviv Polytechnic National University	Lviv	Ukraine
NTU “KhPI”	National Technical University "Kharkiv Polytechnic Institute"	Kharkov	Україна
NTU «KPI»	National Technical University "Igor Sikorsky Kyiv Polytechnic Institute"	Kyiv	Ukraine
NU «OMA»	Національний університет «Одеська морська академія»	Одеса	Україна
NULESU	National University of Life and Environmental Sciences of Ukraine	Kyiv	Ukraine
NUOS	NATIONAL UNIVERSITY OF SHIPBUILDIN NAMED BY ADM. MAKAROV	Nikolaev	Ukraine
ONAFT	Odessa National Academy of Food Technologies	Odessa	Ukraine
ONU	Odessa I.I.Mechnikov National University	Odessa	Ukraine
SSU	Sukhumi State University	Sukhumi	Georgia
VNTU	Vinnytsia National Technical University	Vinnytsia	Ukraine
БНТУ	Белорусский национальный технический университет	Минск	Белоруссия
ВНТУ	Вінницький національний технічний університет	Вінниця	Україна
ДВНЗ «КНУ»	Державний вищий навчальний заклад «Криворізький національний університет»	Кривий Ріг	Україна
ДонНТУ	Донецький національний технічний університет	Покровськ	Україна
ІК НАН України	Інститут кібернетики імені В.М. Глушкова НАН України	Київ	Україна
НТУ «ХПІ»	Национальный технический университет "Харьковский политехнический институт"	Харків	Україна
НТУУ "КПІ"	Національний технічний університет «Київський політехнічний інститут» імені Ігоря Сікорського"	Київ	Україна
НУ «ЛП»	Національний університет «Львівська політехніка»	Львів	Україна
ОДАТРЯ	Одеська державна академія технічного регулювання та якості	Одеса	Україна

Продовження таблиці 1

Скорочення	Повна назва організації	Місто	Країна
ОНАЗ	Одеська національна Академія зв'язку ім. О.С. Попова	Одеса	Україна
ОНАПТ	Одесская национальная академия пищевых технологий	Одесса	Украина
ОНАХТ	Одеська національна академія піщевіх технологій	Одеса	Україна
ОНПУ	Одеський національний політехнічний університет	Одеса	Україна
ОНУ	Одеський національний університет імені І. І. Мечникова	Одеса	Україна
ОТК ОНАХТ	Одеський технічний коледж Одеської національної академії харчових технологій	Одеса	Україна
ПНПУ	Південноукраїнський національний педагогічний університет ім. К.Д. Ушинського	Одеса	Україна
ХНУРЕ	Харківський національний університет радіоелектроніки	Харків	Україна
ХРТК	Харківський радіотехнічний технікум	Харків	Україна
ЦНДІ ОВТ ЗС України	Центральний науково-дослідний інститут озброєння та військової техніки Збройних Сил України	Київ	Україна
ЮНПУ	Южноукраинский национальный педагогический университет им. К.Д.Ушинского	Одесса	Украина

УДК 004.94

IGOR MAZUROK, YEVHEN LEONCHYK, SERHII ORLOV
Odessa I.I.Mechnikov National University (Ukraine)

THE CRYPTOGRAPHIC PROOF-OF-REPLICATION PROTOCOL FOR DISTRIBUTED FILE STORAGE

This paper decides a problem of proving existence given number of unique replicas of the data file which are stored by the nodes of the distributed storage. Usually, proof-of-replication is a proving protocol in which a prover defends a publicly verifiable claim that it is dedicating unique resources to storing several retrievable replicas of a data file. We are considering this problem in the distributed system context of autonomous replicators.

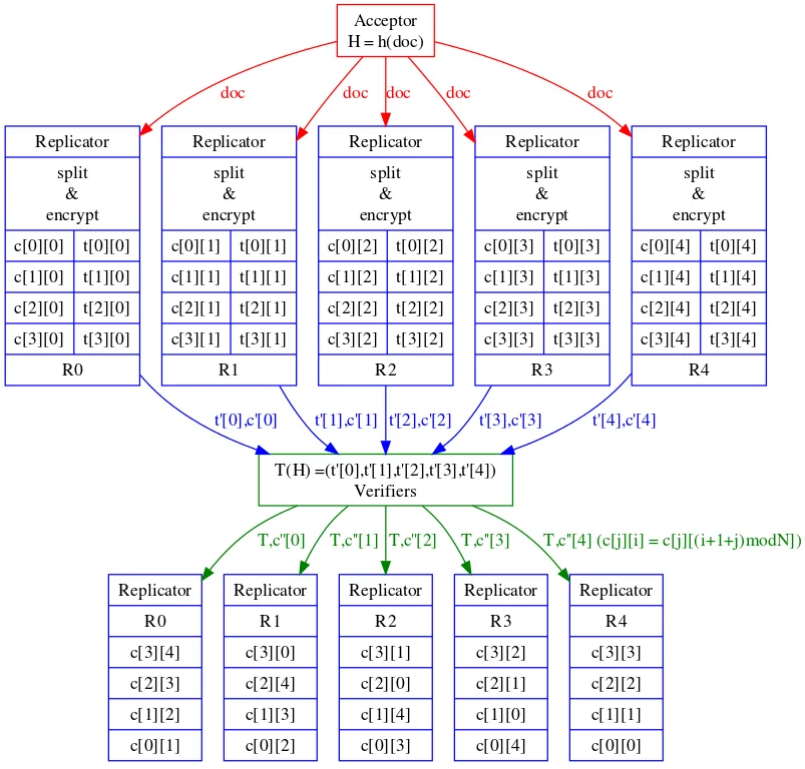
Consider a distributed system of autonomous nodes that must be guaranteed to store their replica of a file [1]. The nodes of such a system will be called replicators. Each node should be able to verify that any other node stores its replica of the file. The node that is currently performing such verification will be called a verifier. A method for generating unique copies, proof of possession of the copy, and proof of the data file possession protocol will be proposed next. Each Replicator plays a Verifier role. We suppose that Storage contract was created before.

All *N* Replicators receive a file for storage via BitSwap protocol (via Client and Replicators that have already downloaded this file). The file hash is in the corresponding transaction and each replicator can make sure it gets the same copy.

Each Replicator splits the file into *N*-1 part.

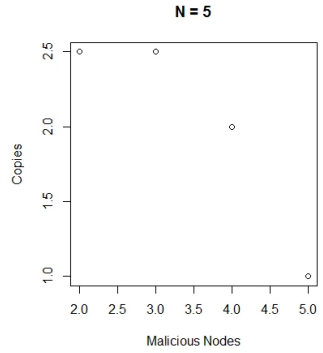
Replicators are sorted by reputation and get their numbers in the contract.

Replicators encode the file parts by their secret key and prepare metadata for verifying (CPOR [2] or another approach).



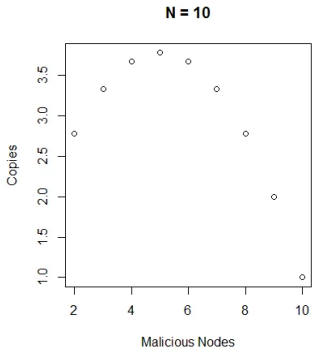
Each Replicator requests the file parts from the other Replicators and sends the file parts with encrypted relevant metadata to the other Replicators. Replicator R_k requests from Replicator R_i the file part number $k - i \bmod N$. R_i sends the j-th part to the Replicator $R_{(i+j) \bmod N}$ responding to the request this node had sent. Thus, each Replicator has all parts of the file but each encrypted by own secure key. Such an approach does not allow to generate information for storage by itself. Here file streaming is not paid. Each Replicator can open and check received parts with metadata. If all is OK and Replicator agrees to be checked with the metadata, it encrypts the metadata with its secure key too. Then it sends metadata to all Replicators. They get encrypted metadata with both secure keys and can check this part of the file.

If there is a mismatch between the file part and its metadata, it starts a retransmitting protocol: a Replicator creates a multi-signed transaction to ban the bad Replicator and asks one of the other Replicators (who can sign the multisig transactions) to request needed part of the file from the suspicious node. If it also fails, such Replicator signs the ban multisig transaction and asks another to request that part. If the required number of Verifiers fail to get the part of the file, the bad Replicator is banished by multisig of Replicators. If one of the Replicator succeeds, it sends the data to the initiator of the process.



The Replicators store all the parts sent to them. Every Replicator can verify all Replicators. If there are k_1 Replicators in the first block $(k_1 < N)$, they are sorted by reputation, get numbers from 0 to $k_1 - 1$ and exchange only required parts in their circle waiting for other Replicators. When another block of k_2 Replicators appear (paying the storage deposit to the contract account) they are also sorted by reputation, get their numbers from k_1 to $k_1 + k_2 - 1$ and exchange the required parts with all other Replicators. The Replicators that have got numbers bigger than $N - 1$ are considered as extra nodes.

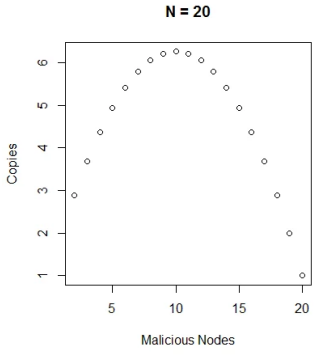
Replacing the outgoing nodes. Let Replicator R_i leaves the group after some time. There are $N-1$ Replicators then and $N-1$ copies of the file, and we need another Replicator at this place. When another Replicator joins the contract it asks all $N-1$ Replicators left for a part of the file and collect the whole file to store. The Replicators send the same parts as they did to the old one.



Now let two Replicators R_i and R_j $(i < j)$ leave the group. A new Replicator takes place i and asks each $N-2$ Replicators left for the part of the file. It collects everything but part that Replicator R_j should send. It encrypts with its own key the part which Replicator R_i should send to Replicator R_j. Instead of part from Replicator R_j it temporary stores nothing waiting for Replicator R_j. When Replicator R_j joins the contract, it collects $N-1$ parts of the file, encrypt the part it should send to Replicator R_i with its own key and sends it to the Replicator R_i.

The same protocol works when there are k Replicators leave. A new Replicator R_i^1 asks $N-k$ Replicators left for a part of the file. It collects everything but parts it should get from Replicators R_i^2, ..., R_i^k. Instead of these parts it temporary stores nothing waiting for the new nodes. When Replicator R_i^2 joins the contract, it collects $N-k+1$ parts of the file, encrypt the appropriate part with his own key and sends this part to the Replicator R_i^1 and so on.

Extra nodes. Now let there are more than N nodes would like to store the file. In any case, there are first initial N Replicators that split and encrypt the file (If there are less than N Replicators at first, they wait till there are at least N for full Cross-verification). The extra Replicators get the parts from initial Replicators. The extra Replicators are also sorted by reputation and get their number in the contract to avoid the situation when extra Replicators store the same selection of parts. The simplest way is that R_N gets part i from R_i and R_{N+k} gets part i from $R_{(i+k) \bmod N}$. So malicious node has to store at least N copies of the file to get the same selection which makes it not beneficial.

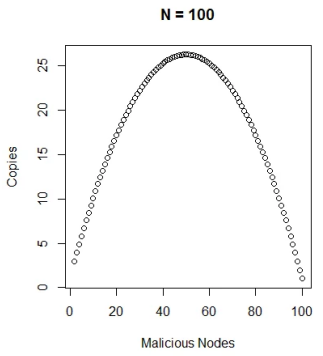


Malicious nodes. Let a group of M malicious nodes exchanged their keys. They want to store one unencrypted copy for all instead of M different copies. But they must store the file parts received from honest nodes too. The next figures show how many different copies such malicious groups must store to pass verification successfully.

For example, let $N = 5$ and $M = 3$. Every malicious node must store 2 parts received from 2 honest nodes. It equals to $2 \times 3 = 6$ parts of the file or 3/2 of the file. And they store one unencrypted copy. So, they equal to 2.5 copies. But if they lose the unencrypted copy, they lose 3 deposits.

Moreover, such behavior is not economically beneficial! There is a legal way to make M deposits and to store one copy. In this way, a node can get the same reward with fewer verifications legally.

Sizing of pieces. It's always more convenient to split the file in the early beginning and distribute its pieces across the system rather than the whole big piece of cake. The same principle works here. The original file is split into blocks size 256 kilobytes (= 262144 bytes). That will provide better performance during encryption and part exchange. Then with each piece of the file, the nodes perform the procedure described above. In this case, it's much more comfortable to split the block for encryption into the number of parts which is a divisor of 262144, the only possible way to split it into whole parts. Also, we need to split the block into parts that are bigger than 32 bytes to get the correct hash. So N is desirable to be 2, 3, 5, 9, 17, 33, 65, ..., $2^k + 1$, ..., 8193.



There are then two possible ways to realize it:

- Let user choose N out of these numbers

- User freely sets the number *N* of desired replicas, but the number of cross-verification parts is the closest desirable number for splitting bigger than the user's set *N*.

The Block size isn`t more than 256kb. The total number of blocks equals $K = \lfloor \frac{FileSize}{256\ Kb} \rfloor$. The number of blocks which will be exactly transmitted by one node equals $M = \lfloor \frac{K}{2N-1} \rfloor$. The rest of the blocks (0, 1, ... , N-2) are transmitted by one per node for which there will be enough blocks. For example, we have 18 blocks and 5 nodes. The first node sends 5 blocks to the second, 5 blocks to the third, 4 blocks to the fourth, 4 blocks to the fifth. Other nodes also do it this way.

The advantage of the proposed approach is the high reliability of storage evidence and the possibility of cross-verification of replicators [3, 4]. The disadvantages of the method depend on the selected asymmetric encryption system - either a low replica preparation speed either an increase of the replica size vise the original data.

References

1. M. Etemad and A. Kupcu, “Transparent, distributed, and replicated dynamic provable data possession.” in Proceedings of the 11th International Conference on Applied Cryptography and Network Security. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 1–18.
2. H. Shacham and B. Waters. “Compact Proofs of Retrievability.” in Journal Cryptology 26(3):442–83, Jul. 2013
3. Yu Chen1, Feng Wang2, Liehuang Zhu1 and Zijian Zhang. “A Survey of Remote Data Integrity Checking: Techniques and Verification Structures.” in International Journal of Grid Distribution Computing Vol. 8, No.4, (2015), pp. 179-198
4. Frederik Armknecht, Ludovic Barman, Jens-Matthias Bohli, Ghassan O. Karame. “Mirror: Enabling Proofs of Data Replication and Retrievability in the Cloud” in the Proceedings of the 25th USENIX Security Symposium, August 10–12, 2016 • Austin, TX, ISBN 978-1-931971-32-4, pp. 1051-1068.

XII МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ І АВТОМАТИЗАЦІЯ – 2019

INFORMATION TECHNOLOGIES AND AUTOMATION– 2019

ОДЕСА
17– 18 ЖОВТНЯ, 2019

Збірник включає доповіді учасників XII Міжнародної науково-практичної конференції «Інформаційні технології і автоматизація – 2019»

Редакційна колегія: Котлик С.В., Хобін В.А., Плотніков В.М.

Комп'ютерний набір і верстка: Соколова О.П.

Відповідальний за випуск: Котлик С.В.