SCIENCE CHINA
Information Sciences

• LETTER •

December 2021, Vol. 64 229301:1–229301:2
https://doi.org/10.1007/s11432-020-2997-0

Secure network coding from secure proof of retrievability

Jinyong CHANG[1], Bilin SHAO[2], Yanyan JI[2*], Maozhi XU[3] & Rui XUE[4]

[1]*School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an* 710055*, China;*
[2]*School of Management, Xi'an University of Architecture and Technology, Xi'an* 710055*, China;*
[3]*School of Mathematical Sciences, Peking University, Beijing* 100871*, China;*
[4]*State Key laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093*, China*

Received 23 March 2020/Revised 20 May 2020/Accepted 6 July 2020/Published online 24 May, 2021

Citation Chang J Y, Shao B L, Ji Y Y, et al. secure network coding from secure proof of retrievability. Sci China Inf Sci, 2021, 64(12): 229301, https://doi.org/10.1007/s11432-020-2997-0

Dear editor,

Compared with the traditional storing-and-forwarding routing mechanism, network coding has become a more attractive paradigm because it has been proven capable of achieving maximized throughput, enhanced robustness, and lower energy consumption for communication networks [1]. Unfortunately, this paradigm is highly susceptible to pollution attack [2]. More specifically, if a packet is modified illegally (i.e., polluted packet), then this modification may quickly spread to the whole network because the intermediate node encodes all received packets including the polluted one [3]. Therefore, it is critical for the intermediate and terminal nodes to check whether a data packet is polluted, which is also the target of secure network coding (NC) scheme [4].

In recent years, storage-as-a-service has emerged as a commercial alternative for user's local data storage due to its features include less initial infrastructure setup, relief from maintenance overhead, and universal access to the data irrespective of the location and devices [5]. However, it also faces several security threats. One of the most serious threats is the integrity of user's stored data. In particular, when storing the data file to a cloud service provider (CSP), a user (or data owner) will delete it from his/her local devices and hence lose local control of it. In this case, CSP may discard some user's rarely accessed data to save its space and earn more profit. Meanwhile, the CSP can lie about the fact. Obviously, it is extremely unfavorable for users. Proof of retrievability (PoR) protocol is just one of initial attempts to formulize the notion of "remotely and reliably checking data's integrity without downloading the whole data file".

In the recent work, Chen et al. [6] revealed a relationship between cloud storage and NC, although these two areas seem to be quite different in their nature and were studied independently. Specifically, they proposed a general transformation from any secure NC scheme to secure PoR protocol. This connection immediately implies many previous secure NC schemes can automatically be used to construct secure PoR protocols. Meanwhile, they also showed that the reverse direction is not correct in general unless imposees special conditions.

Our contribution. In this study, we re-consider the reverse direction proposed by Chen et al. Specifically, we impose constraints on the current notion of PoR protocol and propose a new notion: admissible PoR protocol, which is a special PoR protocol with additional conditions named "Challenge with index-coefficient-form", "Proof with linear-combination-form" and "Proof-aggregation". Then from any secure admissible PoR protocol, we can give a general construction of a secure NC scheme, whose security is based on the underlying PoR protocol.

More concretely, let PoR = ($\mathtt{KeyGen}'$, $\mathtt{Outsource}'$, $\mathtt{Chal}'$, $\mathtt{ProofGen}'$, $\mathtt{Verify}'$) be an admissible PoR protocol. Here, $\mathtt{KeyGen}'$ is an algorithm to generate a user's private key. Taking the data file and user's private key as inputs, $\mathtt{Outsource}'$ generates the authenticated data file, which will be transmitted to CSP for storing. $\mathtt{Chal}'$ outputs the verifier's challenge message, which is given to CSP. The property of "Challenge with index-coefficient-form" requires that the challenged message should be in the form of index-coefficient (j, c_j), where j is the challenged block's position index and c_j is a random coefficient. $\mathtt{ProofGen}'$ is run by CSP and will output a returned proof (denoted by Γ) based on the stored (authenticated) data file and the challenge message. "Proof with linear-combination-form" requires that Γ should have the form of $(\boldsymbol{u}, \sigma)$, in which $\boldsymbol{u}$ is a linear combination of those packets with the indices and coefficients in the challenge message, and σ is the tag of $\boldsymbol{u}$. In addition, the property "Proof-aggregation" refers to that, there exists an efficient aggregation algorithm $\mathtt{Aggr}'$, who can aggregate a group of proof-coefficient pairs (Γ_i, c_i)'s into a new one Γ. Finally, $\mathtt{Verify}'$ is run by the verifier who checks if user's data file is intact. The formal definition of (admissible) PoR protocol can be found in Appendix A.

* Corresponding author (email: yany_ji@163.com)

In general, a secure network coding scheme NC consists of four algorithms `KeyGen`, `Sign`, `Combine`, `Verify`. Here, `KeyGen` is the key-generation algorithm for the source node. `Sign` is also run by the source node to compute the authenticated packets. `Combine` is an algorithm run by intermediate nodes to generate a combined packet of its receiving packets. Finally, `Verify` is run by intermediate or terminal nodes to check the correctness of the received packets. The formal definition of NC scheme can be found in Appendix B.

Now, we describe our general construction of the NC from any admissible protocol PoR. First, `KeyGen` and `Sign` essentially equal to $\texttt{KeyGen}'$ and $\texttt{Outsource}'$, respectively. The algorithm $\texttt{Chal}'$ is run by intermediate nodes to generate the random coefficients of received packets. The two properties of "Proof with linear-combination-form" and "Proof-aggregation" guarantee that the intermediate nodes can correctly generate the combined packets in `Combine`. In addition, `Verify` essentially equals to $\texttt{Verify}'$, in which the property "Challenge with index-coefficient-form" ensures their compatibility. Then we have the following:

Theorem 1. If the admissible PoR protocol PoR is secure, then the scheme NC constructed from PoR is also secure.

The formal transformation and its security proof are presented in Appendix C.

Discussions. To demonstrate the power of our general construction, we present some concrete instantiations. First, we can prove that the transformed PoR protocols (from secure NC schemes) by Chen et al. [6] are admissible and thus can be naturally reversed. Then, we can also obtain other concrete constructions of NC schemes from private-key and public-key PoR protocols like [7, 8] respectively, which are presented in Appendix D, respectively,

Conclusion. We consider the reverse direction of the work in [6]. Concretely, we consider the problem that under what conditions, it is possible to construct a secure network coding scheme from any secure PoR protocol. In fact, by imposing three new constraint conditions on the standard PoR protocol (i.e., admissible PoR), we give a general construction of the NC scheme from the admissible PoR protocol.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 61672416, 61872284, 61772514, 61672059), in part by National Key R&D Program of China (Grant No. 2017YFB1400700), and in part by Beijing Municipal Science & Technology Commission (Grant No. Z191100007119006).

Supporting information Appendixes A–D. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Chang J Y, Shao B L, Ji Y Y, et al. Comment on "a tag encoding scheme against pollution attack to linear network coding". IEEE Trans Parallel Distrib Syst, 2020, 31: 2618–2619
2 Boneh D, Freeman D, Katz J, et al. Signing a linear subspace: signature schemes for network coding. In: Proceedings of International Workshop on Public Key Cryptography, 2009. 68–87
3 Chang J Y, Wang H Q, Wang F, et al. RKA security for identity-based signature scheme. IEEE Access, 2020, 8: 17833–17841
4 Agrawal S, Boneh D. Homomorphic MACs: MAC-based integrity for network coding. In: Proceedings of International Conference on Applied Cryptography and Network Security, 2009. 292–305
5 Ji Y Y, Shao B L, Chang J Y, et al. Privacy-preserving certificateless provable data possession scheme for big data storage on cloud, revisited. Appl Math Comput, 2020, 386: 125478
6 Chen F, Xiang T, Yang Y Y, et al. Secure cloud storage meets with secure network coding. IEEE Trans Comput, 2016, 65: 1936–1948
7 Zhang R, Ma H, Lu Y, et al. Provably secure cloud storage for mobile networks with less computation and smaller overhead. Sci China Inf Sci, 2017, 60: 122104
8 Shacham H, Waters B. Compact proofs of retrievability. J Cryptol, 2013, 26: 442–483