**INTRODUCTION**

WITH the advent of cloud computing, cloud servers offer to their clients (cloud users) various services that include delegation of huge amount of computation and outsourcing large amount of data. For example, a client having a smart phone with a low-performance processor or limited storage cannot accomplish heavy computation or store large volume of data. Under such circumstances, she can delegate her computation/storage to the cloud server. In case of storage outsourcing, the cloud server stores massive data on behalf of its clients (data owners). However, a malicious cloud server can delete some of the client’s data (that are accessed infrequently) to save some space. Secure cloud storage protocols (two-party protocols between the client and the server) provide a mechanism to detect if the server stores the client’s data untampered. Based on the nature of the outsourced data, these protocols are classified as: secure cloud storage protocols for static data (SSCS) [2], [3], [4] and for dynamic data (DSCS) [5], [6], [7], [8]. For static data, the client cannot change her data after the initial outsourcing (e.g., backup/archival data). Dynamic data are more generic in that the client can modify her data as often as needed. In secure cloud storage protocols, the client can audit the outsourced data without accessing the whole data file, and still be able to detect unwanted changes in data done by a malicious server. During an audit, the client sends a random challenge to the server which produces proofs of storage (computed on the stored data) corresponding to that challenge. Secure cloud storage protocols are publicly verifiable if an audit can be performed by any third party auditor (TPA) using public parameters; or privately verifiable if an auditor needs some secret information of the client. The entities involved in a secure cloud storage protocol and the interaction among them are shown in Figure 1. In a network coding protocol [9], [10], each intermediate node (except sender/receiver nodes) on a network path combines incoming packets to output another packet. These protocols enjoy higher throughput, efficiency and scalability than the store-and-forward routing, but they are prone to pollution attacks by malicious intermediate nodes injecting invalid packets. These packets produce more such packets downstream, and the receiver might not finally decode the file sent by the sender node. Secure network coding (SNC) protocols use cryptographic techniques to prevent these attacks: the sender authenticates each packet by attaching a small tag to it. These authentication tags are generated using homomorphic message authentication codes (MACs) [11] or homomorphic signatures [12], [13], [14], [15]. Due to homomorphic property, an intermediate node can combine incoming packets (and their tags) into a packet and its tag. In this work, we look at the problem of constructing a secure cloud storage protocol for dynamic data (DSCS) from a different perspective. We investigate whether we can construct an efficient DSCS protocol using an SNC protocol. In a previous work, Chen et al. [16] reveal a relationship between secure cloud storage and secure network coding. In particular, they show that one can exploit some of the algorithms involved in an SNC protocol in order to construct a secure cloud storage protocol for static data. However, their construction does not handle dynamic data — that makes it insufficient in many applications where a client needs to update (insert, delete or modify) the remote data efficiently. Further investigations are needed towards an efficient DSCS construction using a secure network coding (SNC) protocol. Network coding techniques have been used to construct distributed storage systems [17], [18] where the client’s data are disseminated across multiple servers. However, they primarily aim to reduce the repair bandwidth when some of the servers fail. On the other hand, we explore whether we can exploit the algorithms involved in an SNC protocol to construct an efficient and secure cloud storage protocol for dynamic data (for a single storage server). Although dynamic data are generic in the sense that they support arbitrary update (insertion, deletion and modification) operations, append-only data (where new data corresponding to a data file are inserted only at the end of the file) find numerous applications as well. These applications primarily maintain archival as well as current data by appending the current data to the existing datasets. Examples of append-only data include data obtained from CCTV cameras, ledgers containing monetary transactions, medical history of patients, data stored at append-only databases, and so on. Append-only data are also useful for maintaining other log structures (e.g., certificates are stored using append-only log structures in certificate transparency schemes [39]). In many of such applications, the data owner requires a cloud server to store the bulk data in an untampered and retrievable fashion with append being the only permissible update. Although secure cloud storage schemes for generic dynamic data also work for append-only data, a more efficient solution (specific to append-only data files) would be helpful in this scenario. Our Contribution: Our major contributions in this work are summarized as follows. • We explore the possibility of providing a generic construction of a DSCS protocol from any SNC protocol. We discuss the challenges for a generic construction in details and identify some SNC protocols suitable for constructing efficient DSCS protocols. • We construct a publicly verifiable DSCS protocol (DSCS I) from an SNC protocol [15]. DSCS I handles dynamic data, i.e., a client can efficiently perform updates (insertion, deletion and modification) on the outsourced data. We discuss the (asymptotic) performance and certain limitations of DSCS I. • We provide the formal security definition of a DSCS protocol and prove the security of DSCS I. • As append-only data are a special case of generic dynamic data, we can use DSCS I (which is based on [15]) for append-only data. However, we identify some SNC protocols that are not suitable for building a secure cloud storage for generic dynamic data, but efficient secure cloud storage protocols for appendonly data can be constructed from them. We construct such a publicly verifiable secure cloud storage protocol (DSCS II) for append-only data by using an SNC protocol proposed by Boneh et al. [13]. • We discuss the (asymptotic) performance of DSCS II which overcomes some limitations of DSCS I. • We implement DSCS I and DSCS II and evaluate their performance based on storage overhead, computational cost and communication cost.