**PROTECTING CLOUD DATA WITH DYNAMIC SECURITY VIA NETWORK CODING**

**Abstract:**

In the age of cloud computing, cloud users with limited storage can outsource their data to remote servers. These servers, in lieu of monetary benefits, offer retrievability of their clients’ data at any point of time. Secure cloud storage protocols enable a client to check integrity of outsourced data. In this work, we explore the possibility of constructing a secure cloud storage for dynamic data by leveraging the algorithms involved in secure network coding. We show that some of the secure network coding schemes can be used to construct efficient secure cloud storage protocols for dynamic data, and we construct such a protocol (DSCS I) based on a secure network coding protocol. To the best of our knowledge, DSCS I is the first secure cloud storage protocol for dynamic data constructed using secure network coding techniques which is secure in the standard model. Although generic dynamic data support arbitrary insertions, deletions and modifications, append-only data find numerous applications in the real world. We construct another secure cloud storage protocol (DSCS II) specific to append-only data — that overcomes some limitations of DSCS I. Finally, we provide prototype implementations for DSCS I and DSCS II in order to evaluate their performance.

Index Terms—Secure cloud storage, network coding, dynamic data, append-only data, public verifiability.

**Existing System:**

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 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).

Disadvantages:

1. 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. Secure cloud storage protocols enable a client to check integrity of outsourced data.
2. For static data, the client cannot change her data after the initial outsourcing (e.g., backup/archival data)

**Proposed System:**

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 . 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 ) for append-only data. However, we identify some SNC protocols that are not suitable for building a secure cloud storage for generic dynamic data, butefficient 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..

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.

Advantages:

1. Secure network coding (SNC) protocols use cryptographic techniques to prevent these attacks: the sender authenticates each packet by attaching a small tag to it.
2. 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.

**System Architecture:**



**MODULES:**

**1.Data Owner**

**2.TPA(Trusted Third Party)**

**3.Cloud Server**

**1.Data Owner**

In this application the owner is one of the main module for uploading the files and view the uploads file which are uploaded by the owner before do all these operations the owner should register with the application and the owner should authorized by the cloud.

**2. TPA(Trusted Third Party)**

The TPA is used to generate the Auditing Task for the requested users.Here the trapdoor should login directly with the application

**3.Cloud Server**

The cloud is the main module to operate this project in the users activations, owner activation and also the cloud can check the following operations like search permission provides to the users, can check the top-k searched keyword, top-k similarity in chart, top-k searched keyword in chart. Primarily the cloud should login. Then only the cloud can perform the abovementioned actions.

**SYSTEM REQUIREMENTS:**

**HARDWARE REQUIREMENTS:**

* PROCESSOR :I3.
* Hard Disk : 40 GB.
* Ram : 2 GB.

**SOFTWARE REQUIREMENTS:**

* Operating system : Windows.
* Coding Language : JAVA/J2EE
* Data Base : MYSQL
* IDE :Netbeans8.1