Introduction:  
Most of the cloud service providers like Amazon S3,Microsoft Azure or Google cloud provide SLA of availability of services for 99.95% of the time however this single properties of SLA is not convincing enough to opt for cloud storage. Cloud is un-trusted entity and considered as potential for risks like data leakage, being hacked or returning manipulated data. Our goal is to introduce verifiable security properties in SLA like Confidentiality, Integrity, Write serializability , read freshness. In case of SLA breach, agreed amount can be compensated. We are trying to model a mechanism which will provide proofs of security guarantees and violations if occured. These proofs will be proofs by client as well as cloud to either prove violation or to counter the false accusations with proofs.

Target Scenario:  
We are targeting our model for home user to mainly enterprises which have privacy concerns for hosting their storage to cloud. Our model comprises of 3 roles as shown below.  
1. (Data) owner: the entity who purchases the cloud storage service. A data owner might be an enterprise with business data or a home user with personal data.

2. Cloud: the cloud storage provider.

3. (Data) users: users who are given either read or write access to data on the cloud. A user might be an employee of an enterprise or family members and friends of a home user.  
  
 Role of owner would be to carry auditing of security properties and controlling access permissions for users to the cloud data.

Concept and understanding:

We are aiming to achieve four major security properties in our model and they are defined below.

1. Confidentiality (C) holds when the cloud or any illegitimate user cannot identify the contents of any blocks stored on the cloud.
2. Integrity (I) holds when each read returns the content put by a legitimate user. For example, the cloud cannot replace some data with junk.
3. Write-serializability (W) holds when each user committing an update is aware of the latest committed update to the same block.
4. Freshness (F) holds if reads return the data from the latest committed write.

In our model, we suppose that cloud is hosting blocks of data. Each data block can be read and written by users. In primitive approach every block has its secret key which can be only accessed by legitimate users. This secret key is used for en(de)cryption of the data. This way we achieve confidentiality of the data. This approach can be extended to make families of blocks hence using one single key for the block family and managing permissions for block family rather than per block. This can simplify management and overhead of number of secret keys.  
Every read and write of data involves two parties i.e cloud and user. Owner doesn’t need to be available for any read write requests. Owner performs two functions i.e controlling access permissions and auditing the cloud for any SLA violations.   
In our model during each read and write cloud and client exchange some proofs for requests. These proofs are sent to owner who does auditing later on. Owner can do auditing in real time however this will require owner to be available all the time and need more processing and compute power.   
In our model, cloud is considered as totally untrusted party who can return arbitrary data or it might not honor access control as instructed by owner or there could be bugs in the cloud implementation resulting in vague behavior with data. Our model will make sure that auditing performs security checks and in case of any violation, owner must be able to prove the violation which can not be denied by cloud. Similarly if owner is falsely accusing the cloud then cloud should be able to provide proofs of false accusations.

Major Goals:

* Detection of integrity, write serializability and read freshness. User has to make sure that he is achieving confidentiality by encrypting the data. Cloud can not guarantee confidentiality if user does not encrypt the traffic.
* Cloud violation should be proved whenever they happen. A cloud can be convinced if violation really occurred or cloud will not be convinced if accusation of violation is false.
* Access control mechanism will be scalable since we are targeting from small home users to enterprise size which can have hundreds of thousands of users.
* Security always comes with a cost and our goal is to minimize the overhead to the possible extent without compromising on security requirements.
* We will measure relevant overhead in terms of storage as well as latency.