Official Arbitration Protocol

Following changes are required regarding file, metadata, proof, and update objects:

OAMetadata.h*: Types of 'metadata' and 'proof' data members should be changed. This would also lead the requirement to change the types of two parameters of the constructor, and the return types of two corresponding getters.

OAUpdate.h**: Type of 'update' data member should be changed. This would also lead the requirement to change the type of the first parameter of the constructor, and the return type of the corresponding getter.

OAClient.sendFile(...): file should be changed to an instance of a desired type. And its pointer is to be returned. Therefore, the return type in OAClient.cpp and OAClient.h should change, together with the type of the variable that its return value assigned in Test.cpp at step3.

OAServer.sendMetadata(file)***: Having the type of file changed, the corresponding parameter of this function should also be changed in both OAServer.h and OAServer.cpp. Also, a metadata and its proof should be generated in the function, proof should be provided to the signature generation(instead of the temporary string 'proof'), and all three should be sent to the constructor of OAMetadata.

OAClient.sendPaymentEscrow(...): Client should check the metadata and proof which comes in OAMetadata instance at the beginning of the function in OAClient.cpp. Checking for signature is already coded.

OAClient.update(...): An update instance of a desired type should be generated at the beginning, and it should be sent to the constructor of OAUpdate (instead of the temporary string 'update')

OAServer.sendMetadata(update)***: A metadata and its proof should be generated in the function, proof should be provided to the signature generation(instead of the temporary string 'proof'), and all three should be sent to the constructor of OAMetadata.

OAJudge.checkMetadata(...): Judge should check the metadata and proof which comes in OAMetadata instance at the beginning of the function in OAJudge.cpp. Checking for signature is already coded.

OAServer.sendProof(): A proof showing that the server did not corrupt the client's file should be generated and sent. Return type of the function should be changed in OAServer.h and OAServer.cpp files. It is also required to change the type of the variable that its return value assigned in Test.cpp at OA_CORRUPTION.

OAJudge.checkProof(...): Having the type of 'proof' changed, type of the parameter of this functions should also be changed. And lastly, proof should be checked by the judge.

After the corrections are done, the warnings in the output should be removed. There are 7 warnings reminding the changes to be done. They are printed in the Test.cpp: 2 of them are in the if-block OA CORRUPTION, 3 are in OA_DENIAL_OF_UPDATE, 1 at step12, and last one in step13.

* OAMetadata class is not a metadata type. It is a wrapper for metadata, proof, and signature.

** OAUpdate class is not an update type. It is a wrapper for update and signature.

*** OAServer class has two different member functions sharing the same name, 'sendMetadata'. They can be seperated by checking the parameters.

Signatures are kept as strings. Sizes of the signature instances are around 46-47 bytes. If we try saveGZString(signature), it gives us 105-106 bytes. What's prefered to be used in the following calculations is the result of saveGZString(signature.c_str()) which is 55.

Table 1: **Details of Setup Operations of Official Arbitration Protocol**

Operation	Time(ms)	Size(B)
Signature key generation x2	2x 0.1961	-
Public key generation and transfer x2	2x 0.0201	2x 521
Server sends signature on proof of metadata	0.2049	55
Client checks signature and sends VE of payment	399.7200	48277
Server checks VE of payment and sends VE of warranty	553.8610	48307
Client checks VE of warranty and sends signature on counter	163.4200	55
Server checks signature and sends signature on counter/metadata	0.3985	55
Server sends receipt	0.2198	55
Client checks receipt and sends endorsement of payment	0.4229	128
Server checks endorsement	0.4984	-
Total overhead of setup for client	563.7791	48981
Total overhead of setup for server	555.3988	48993
Total overhead of setup for client (excluding warranty)	400.5691	48981
Total overhead of setup for server (excluding warranty)	155.8931	686

Table 2: **Details of Update Operations of Official Arbitration Protocol**

Operation	Time(ms)	Size(B)
Client sends signature on counter	0.2042	55
Server checks signature and sends signature on metadata	0.3813	55
Client checks signature	0.2143	-
Total overhead of update for client	0.4185	55
Total overhead of update for server	0.3813	55

Table 3: **Details of Arbitration Operations of Official Arbitration Protocol**

Arbitration setup		
Operation	Time(ms)	Size(B)
Client sends the claim	0.0007	49215
Client sends the claim (without warranty)	0.0007	824
Judge checks the claim	0.7059	-
Server sends counter and signature	0.0009	55
Judge checks signature and compares counters	0.2089	-
Overhead of arbitration setup for client	0.0007	49215
Overhead of arbitration setup for client (without warranty)	0.0007	824
Overhead of arbitration setup for server	0.0009	55
Overhead of arbitration setup for judge	0.9148	-
Server Found Guilty		
Judge decrypts and sends endorsement of warranty	20.6351	128
Client checks the endorsement	0.4984	-
Overhead of server found guilty for client	0.4984	-
Overhead of server found guilty for server	-	-
Overhead of server found guilty for judge	20.6351	128

# of commits	11413	27534	24054	25000
Network overhead per commit (B) (with warranty)	118.58	113.56	114.07	113.92
Network overhead per commit (B) (without warranty)	114.35	111.80	112.06	111.99
Computation overhead per commit (ms) (with warranty)	0.8979	0.8405	0.8463	0.8446
Computation overhead per commit (ms) (without warranty)	0.8486	0.8200	0.8229	0.8221

Table 4: Service overhead, in total for both client and server