CTng Client

Client- UpdateMonitor API core

1) The client will query the monitor for the update, it will send the last update time to the monitor
2) If the UpdateMonitor is not responsive or it has been convicted, go to another monitor for updates
3) The monitor will compute the client update based on the current period and the last update period provided by the client
4) The monitor will then send the Clientupdate Object to the client

Clientupdate Data structure

```
type Clientupdate struct{
    STHs *gossip.Gossip_Storage
    REVs *gossip.Gossip_Storage
    PoMs *gossip.Gossip_Storage
    MonitorID string
    //Period here means the update period, the client udpate object can
contain more information than just the period
    Period string
    PoMsig string
}
```

Note:

1) Gossip storage, STH, REV, PoM specifications are listed below (note STH, REV, and PoM might refers to other type of gossip objects but for the client update, they are referred to STH_FULL, REV_FULL and CONFLICT_POMs respectively)
2) STHs, and REVs are Threshold signed by the monitor-gossiper system, the scheme we used is BLS, each signature corresponds to one STH/REV
3) The PoMsig in this data structure is the monitor's signature over the entire PoMs. A single PoM does not have a signature over it

Gossip_Storage:

Data structure: Map[GossipID]GossipObject
Entity using the storage: Monitor and gossiper
Purpose: Store gossip objects by their ID, ID specified below

GossipID: The Key

Period	string
Type	string
Entity URL	string

Gossip Object

Application	String
Type	String
Period	String
Signer	String
Signers	Map[int]String
Timestamp	UTC-RFC 3339 String
Signature	[2]String
Crypto_Scheme	String
Payload	[3]String

Type 6: STH_FULL (301)

The STH_FULL (STH with Aggregated Signature from m monitors) gossip object is generated by the gossiper once it receives the Threshold number of STH_FRAG, and successfully combines the Signature by invoking the Threshold Aggregate Function. The STH_FULL will then be sent to the monitor for storage purposes. It will also be gossiped to its connected gossipers.

Application	“CTng”
Type	STH_FULL
Period	String
Signer	“”
Signers	List of monitors (only if the Crypto Scheme requires it for verification)
Timestamp	UTC-RFC 3339 String
Signature	Aggregated Threshold Signature
Crypto_Scheme	Threshold Signature Scheme, BLS in our case
Payload	Logger URL, STH (contains timestamp, roothash, tree size)

Type 7:REV_FULL (302)

The REV_FULL (Revocation Information with Aggregated TSS signature from m monitors) is generated by the gossiper once it receives m STH_FRAG, and successfully combines the Signature by invoking the Threshold Aggregate Function. The STH_FULL will then be sent to the monitor for storage purposes. It will also be gossiped to its connected gossipers.

Application	“CTng”
Type	REV_FULL
Period	String
Signer	“”
Signers	List of monitors (only if the Crypto Scheme requires it for verification)
Timestamp	UTC-RFC 3339 String
Signature	Aggregated Threshold Signature
Crypto_Scheme	Threshold Signature Scheme, BLS in our Case
Payload	CA URL, Revocation type (CRV scheme), Revocation Information (SRH and dCRV)

Type 9: CONFLICT_POM (304):

The CONFLICT_PoM is a special type of PoM that does not use the Threshold Signature Scheme (TSS). It is generated by the gossiper upon detecting conflicting information from a monitor/logger/CA. After generation, it will be sent to its own monitor for storage and other gossipers. The structure of the gossip object is listed below:

Application	“CTng”
Type	CONFLICT_POM
Period	String
Signer	“”
Timestamp	UTC-RFC 3339 String
Signature	2 different valid signatures from the same signer
Crypto_Scheme	RSA if the rogue entity is logger or CA BLS if the rogue entity is Monitor
Payload	Entity URL this CONFLICT_POM is against, Gossip_object_1_payload, Gossip_object_2_payload (Gossip_object_1 and Gossip_object 2 have 2 different hashes/ same period numbers/ same object type)

Client-CheckMonitor API core

1) The client will send PoMs, Period, and PoMsig it receives from the UpdateMonitor to the check monitor Periodically
2) The check monitor will compare the received PoMs with its own PoM storage
3) If there is a discrepancy, the CheckMonitor will initiate an accusation against the Update Monitor and then send the PoMs, Period, and PoMsig to all the monitors in the network.
4) A PoM against the update monitor will be generated if the number of accusations reaches the Threshold
5) The Check Monitor will then send the PoM (against the UpdateMonitor) to the client
6) The UpdateMonitor will no longer be trusted

Client-Subject API core

1) The subject will send certificate + STH + POI to the client
2) The client can then verify if the certificate is in the STH given the POI
3) Alternatively, the client can get the REV from the Subject if the client cannot connect to the UpdateMonitor
4) The client can verify the dCRV given the SRH

CTng Client function overall