Let P ($|P| = t$) be a group of shareholders which have not been detected to be cheating when issuing the certificate. Then we have

$$\begin{aligned}
SK_uG &\overset{(5)}{=} (c_u + \sum_{i \in P} \gamma_i \omega_i)G \\
&\overset{(3)}{=} c_uG + \left(\sum_{i \in P} (\beta_i + h(I_u, C)\alpha_i) \right) G\omega_i \\
&= V_u + \sum_{i \in P} (\beta_i \omega_i G + \alpha_i \omega_i h(I_u, C)G) \\
&= V_u + V + h(I_u, C)PK_0 \\
&= C + h(I_u, C)PK_0 \quad \Box
\end{aligned}$$

8.2 Detectability

We have to verify that every shareholder not following the protocol will be detected.

Key Generation During key generation, we use the protocol described in [11]. This protocol has already been proven to be robust, i.e., players not following the protocol will be detected.

Certificate Issuing First, the players generate a distributed secret with Pedersen's protocol (which is proved to be detectable). Second, they reveal $\{\gamma_i\}$, but these values are verified through equation (4). Finally, they send the calculated certificate to the user. By verifying equation (6), the user can identify the correct certificates.

8.3 Notion of Security in the Random Oracle Model

We assume that we are in the random oracle model (i.e., the hash function is modelled as a random function; see [1]). Let (SK_{CA}, PK_{CA}) be the key pair of the CA (represented through shareholders in case of the distributed implicit certificate scheme). An implicit certificate scheme is *secure* if the following two properties hold:

unforgeability It is hard for an adversary who does not know CA's secret key to forge implicit certificates in such a manner that the adversary knows the corresponding private key

non-impersonating It is hard for CA to obtain the requester's private key provided that the requester followed the protocol.

The term "hard" means that there is no polynomial-time adversary who can solve the task with non-negligible probability. These conditions must hold for adversaries defined as follows.

We define a forging adversary A_f as a probabilistic, polynomial-time turing machine which, on input PK_{CA} does the following:

- it may watch other entities requesting and receiving implicit certificates from the CA