(2) This scheme is robust, i.e., a corrupt signer who does not follow the protocol (by distributing inconsistent shares) will be detected. The random shared secret protocol has been proven to be robust in [11]. The validity of the $\{\gamma_i\}$ is verified at step 3.

(3) The scheme can easily be modified so that a *trusted combiner* calculates the signature, instead of the players. The γ_i's would be sent secretly to the trusted combiner, who proceeds with the verification and the signature generation. In such a scenario, the players would not be able to generate a signature without the combiner.

(4) The only property required by the underlying secret sharing scheme is that it must be homomorphic. This signature scheme could therefore be generalized to non-threshold access structures by using a suitable linear general access structure secret sharing scheme.

5 Security

5.1 Notion of Security

In this section, we show that the proposed (t, n) threshold signature scheme is as secure as Schnorr's signature scheme, i.e., existentially unforgeable under adaptively chosen message attacks in the random oracle model.

We define an adaptively chosen message attack against our (t, n) threshold scheme as follows. An adversary $A_{DistSchnorr}$ is allowed to have the signature issuing protocol executed by any t or more signers to compute signatures on messages of his own choice. He also might corrupt up to $t-1$ arbitrary players. $A_{DistSchnorr}$ then tries to forge a new signature from the signatures he obtained in this way and from his view, where the *view* is everything that $A_{DistSchnorr}$ sees in executing the key generation protocol and the signature issuing protocol.

Let $A_{NormSchnorr}$ be a successful adversary that can break (in the sense of an existential forgery under adaptively chosen message attack) Schnorr's scheme (denoted by $D_{NormSchnorr}$); and let $A_{DistSchnorr}$ be a successful adversary that can break the distributed Schnorr scheme (denoted by $D_{DistSchnorr}$) presented in this paper. To proof the security of our scheme, we will show that given $A_{NormSchnorr}$, one can construct an adversary $A_{DistSchnorr}$, and visa versa. This implies that $D_{DistSchnorr}$ is as secure as $D_{NormSchnorr}$ is.

The basic idea of how to construct $A_{NormSchnorr}$ given the adversary $A_{DistSchnorr}$, a public key Y and a signing oracle goes as follows. $A_{NormSchnorr}$ simulates the roles of the uncorrupted players during all stages of $D_{DistSchnorr}$ – i.e., from the key generation protocol that outputs Y up to the signature issuing protocols for $A_{DistSchnorr}$'s chosen message attack – and lets them interact with $A_{DistSchnorr}$ (see Section 5.3). Because $A_{DistSchnorr}$ cannot distinguish what he sees (i.e., his view) during this simulation from what he would see during a real run of $D_{DistSchnorr}$, he will succeed and output a valid forgery, and therefore so will $A_{NormSchnorr}$.

The next section explains precisely what a view is. We also explain how to build a simulator *SIM* that simulates the honest players during the generation of a distributed random shared secret such that it produces for an arbitrary but given public key Y a view that is indistinguishable for the adversary from a view that would have resulted from real players during a real run of the same protocol outputting Y. This simulator is then used later as a subroutine of a simulator for the adversary's entire view of our threshold signature scheme.