Not sure if this idea already already exists under another name, so I'll call it "Fiat-Shamir statistical threshold signatures" for the moment.

Suppose that you have N users with N public keys (think: N > 1000). You make a shared public key out of a Merkle root of the public keys. Then, any M of the N users can make a construction as follows:

1. The M users all sign some message, H.
2. A Merkle tree of the signatures is made, where the index of each signature in the tree is the same as the index of its corresponding public key in the public key tree (positions in the tree without a signature can be filled with empty data). The Merkle root of the signature tree (possibly with some proof of work grinding) is used as a source of random entropy, which chooses a random k

of the leaves of the tree.

1. Any d

of these k

leaves (think: d/k ~= M/N

), together with the Merkle branches proving that they come from the Merkle root, together with the Merkle branches proving the public keys associated with those signatures are actually part of the root public key, can be used as a "signature" to statistically prove that close to M valid signatures for H were actually part of the Merkle tree.

As k

increases, this scheme becomes statistically more accurate, and comes close to taking the role of a threshold signature scheme. It also has the benefits that:

- It is completely independent of the type of underlying signature used.
- It even allows the underlying signature schemes to be different; each public key could be validation code for some arbitrary signature verification function.
- It is thus (potentially) quantum-resistant.
- It does not require any kind of distributed key exchange.
- Complexity is linear in the number of participants.
- Anyone who participated in a signature, even those who were not included in the random sample, can later prove that they did so.

Its main disadvantages are:

- Large size (32 * k * log(N) + 32 * d * log(N) + d * signature_length)
- It does not generate a unique and manipulation-resistant random value as a byproduct.