A Survey on Distributed Randomness

Kevin Choi

Spring 2021

Contents

1 Building Blocks

1.1 Commit-Reveal

In the case of a strawman solution, the classical commit-reveal provides an intuitive way to source a random number from a group of nodes by first allowing each node to commit (in the cryptographic sense) to a secret random number and then adding all shares (revealed later) from all nodes to compute the final random number of a round. The problem with this is that the last person to reveal his share can in fact check the round's output faster than others and hence can decide not to reveal his number if he doesn't like the round's output, biasing the distributed randomness. This is called the *last revealer attack*.

1.2 Verifiable Secret Sharing

The issue with the classical Shamir's secret sharing scheme is that either the dealer or other participants could in fact be acting maliciously, e.g. Alice (a participant) needs to trust that she has received her share correctly from the dealer while she also needs to trust that other participants' revealed shares are correct.

This issue can be fixed by the notion of verifiable secret sharing (VSS). The idea is that we add an additional verification process to the usual Shamir's secret sharing scheme. Here, we take note of two commonly used VSS schemes: Feldman-VSS and Pedersen-VSS.

The mathematical setup is as follows. We choose primes p and q such that $q \mid p-1$ and let g be a generator of G_q, a cyclic subgroup of $\mathbb{Z}_p^*$. This setup has the effect of achieving the following: $a \equiv b \pmod{q} \iff g^a \equiv g^b \pmod{p}$. As a result, it should be understood that modular arithmetics are done in modulo q whenever the numbers involved concern the exponents while in modulo p otherwise. For convenience, we may omit mod p such that one can aptly assume arithmetics are done in modulo p given an equation unless stated otherwise.

1.2.1 Feldman-VSS

The following summarizes a simple VSS scheme (where t among n participants can reconstruct the group secret) proposed by Paul Feldman.

- $f(x) = \sum_{i=0}^{t-1} a_i x^i$ is randomly selected by the dealer, where $a_i \in \mathbb{Z}_q$ and $f(0) = a_0$ is the secret
- The shares are $f(1), f(2), ..., f(n)$ in mod q and are distributed to n participants, respectively
- Also distributed from the dealer are commitments to coefficients of f, i.e. $c_j = g^{a_j}$ for $j = 0, ..., t-1$

- Given her share $f(k)$ and the polynomial coefficient commitments, Alice (a participant) can verify her share by checking:

$$g^{f(k)} = g^{\sum_{i=0}^{t-1} a_i k^i} = \prod_{j=0}^{t-1} c_j^{k^j} = c_0 c_1^k c_2^{k^2} \cdots c_{t-1}^{k^{t-1}}$$

- Any t number of participants (say) $i = 1, 2, ..., t$ can recover the secret a_0 by performing Lagrange interpolation involving Lagrange coefficients $\lambda_i = \prod_{j \neq i} \frac{j}{j-i}$ in mod q:

$$a_0 = f(0) = \sum_{i=1}^{t} f(i)\lambda_i$$

What is new here (compared to Shamir's secret sharing scheme) is the inclusion of commitments to polynomial coefficients in the scheme. These commitments enable participants to verify the validity of their corresponding shares.

1.2.2 Pedersen-VSS

Reminiscent of Pedersen commitment, Pedersen-VSS is a variation that involves two random polynomials generated by the dealer as opposed to one. Namely, the scheme runs as follows.

- $f(x) = \sum_{i=0}^{t-1} a_i x^i$ and $f'(x) = \sum_{i=0}^{t-1} b_i x^i$ are randomly selected by the dealer, where $a_i, b_i \in \mathbb{Z}_q$ and $f(0) = a_0$ is the secret (as before)
- The shares are $(f(1), f'(1)), ..., (f(n), f'(n))$ in mod q and are distributed to n participants, respectively
- Also distributed from the dealer are commitments to coefficients of f and f', i.e. $c_j = g^{a_j} h^{b_j}$ for $j = 0, ..., t-1$
- Given her share $(f(k), f'(k))$ and the polynomial coefficient commitments, Alice (a participant) can verify her share by checking:

$$g^{f(k)} h^{f'(k)} = \prod_{j=0}^{t-1} c_j^{k^j}$$

- Any t number of participants (say) $i = 1, 2, ..., t$ can recover the secret a_0 by performing Lagrange interpolation involving Lagrange coefficients $\lambda_i = \prod_{j \neq i} \frac{j}{j-i}$ in mod q:

$$a_0 = f(0) = \sum_{i=1}^{t} f(i)\lambda_i$$

Effectively, what Pedersen-VSS is able to achieve is the decoupling of g^{a_0} (as the public key corresponding to the secret key a_0) and $g^{a_0}h^{b_0}$ (as the published commitment for verification purposes). In other words, the verification process in which the participants verify their shares does not (even information-theoretically) leak any information regarding the initial secret a_0, a fact that is not true with Feldman-VSS.

1.3 Distributed Key Generation

The motivation for distributed key generation (DKG), which basically comprises n parallel instances of a VSS (run by each participant), is to achieve and utilize a group secret in a leaderless manner such that any threshold t number of participants should be able to recover the same group secret. The basic idea is that each participant becomes a dealer (i.e. leader) for a VSS, in which case the protocol deals with n random polynomials in mod q $\{f_i\}_{i=1,\dots,n}$ as opposed to one. Though not computed explicitly, the implicit group polynomial $f = \sum f_i$ then embeds the group secret in the form of $f(0)$ (denoted by x) as well as the corresponding public key $g^{f(0)}$ (denoted by y) via commitments to coefficients of f_i as per each VSS. Notably, this facet is the one that allows a leaderless configuration where there does not exist a leader for f even if each participant P_i remains a leader for f_i with the knowledge of $f_i(0)$. As a result, it is only collectively (i.e. with the collaboration of at least t number of nodes in a group) that the group can make use of x and y in the typical sense of asymmetric cryptography (e.g. signing a message) after performing a DKG.

Before delineating the process of a DKG, we include what it means for a DKG to be uniformly secure. Namely, a uniformly secure DKG should satisfy the following three correctness properties and one secrecy property.

Definition 1.1. In the setting where at most $t-1$ nodes can be controlled by an attacker without compromising the protocol, a DKG is *uniformly secure* if the following properties are satisfied.

Correctness

1. Any subset of shares of size t can be used to recover the same group secret key x.

2. All honest parties have access to the same public key $y = g^x$.

3. x is uniformly distributed in $\mathbb{Z}_q$, and thus y is uniformly distributed in G_q (subgroup of $\mathbb{Z}_p^*$ generated by g).

Secrecy

1. Other than from the fact that $y = g^x$, no information on x is leaked. Equivalently, this can be stated using a simulator SIM: for every probabilistic

polynomial-time adversary $\mathcal{A}$, there exists a probabilistic polynomial-time simulator SIM that, given $y \in G_q$ as input, produces an output distribution which is polynomially indistinguishable from $\mathcal{A}$'s view of a DKG protocol that ends with y as its public key output while $\mathcal{A}$ is allowed to corrupt up to $t-1$ participants.

The third correctness property is the one that motivates the ....
As the goal in practical terms is to use

1.3.1 Joint-Feldman

1. Each participant P_i runs a Feldman-VSS by choosing a random polynomial $f_i(z) = \sum_{j=0}^{t-1} a_{ij} z^j$ and sending a "subshare" $f_i(j)$ to player P_j for all j

2. To satisfy the verifiability portion of the VSS, P_i broadcasts $A_{ik} = g^{a_{ik}}$

3. Upon receiving the subshares and the corresponding commitments (e.g. in the form of a verification vector), P_j can use the verification mechanism per VSS to verify the subshares. If a verification fails, P_j can broadcast a complaint against P_i.

4. If P_i receives at least t complaints, then P_i is disqualified. Otherwise, P_i needs to reveal the subshare $f_i(j)$ per P_j that has broadcasted a complaint. We call $QUAL$ the set of non-disqualified players.

5. Once $QUAL$ is set, we define $f(z) = \sum_{i \in QUAL} f_i(z) = \sum_{i=0}^{t-1} a_i z^i$ such that each participant P_j in $QUAL$ can compute the group public key $y = g^{f(0)} = \prod_{i \in QUAL} A_{i0}$, commitments to f's coefficients $A_k = g^{a_k} = \prod_{i \in QUAL} A_{ik}$, and P_j's share (of the group secret) from subshares $f(j) = \sum_{i \in QUAL} f_i(j)$. Though not computed explicitly, the group secret key x is then equal to both $\sum_{i \in QUAL} a_{i0}$ and the Lagrange interpolation involving the shares $\{f(j)\}_{j \in QUAL}$.

1.3.2 Joint-Pedersen

1. Each participant P_i runs a Pedersen-VSS by choosing two random polynomials $f_i(z) = \sum_{j=0}^{t-1} a_{ij} z^j$ and $f'_i(z) = \sum_{j=0}^{t-1} b_{ij} z^j$ and sending a "subshare" $(f_i(j), f'_i(j))$ to player P_j for all j

2. To satisfy the verifiability portion of the VSS, P_i broadcasts $C_{ik} = g^{a_{ik}} h^{b_{ik}}$

3. Upon receiving the subshares and the corresponding commitments (e.g. in the form of a verification vector), P_j can use the verification mechanism per VSS to verify the subshares. If a verification fails, P_j can broadcast a complaint against P_i.

4. If P_i receives at least t complaints, then P_i is disqualified. Otherwise, P_i needs to reveal the subshare $(f_i(j), f'_i(j))$ per P_j that has broadcasted a complaint. We call $QUAL$ the set of non-disqualified players.

5. Once $QUAL$ is set, we define $f(z) = \sum_{i \in QUAL} f_i(z) = \sum_{i=0}^{t-1} a_i z^i$ such that each participant P_j in $QUAL$ can compute the group public key $y = g^{f(0)} = \prod_{i \in QUAL} A_{i0}$, commitments to f's coefficients $A_k = g^{a_k} = \prod_{i \in QUAL} A_{ik}$, and P_j's share (of the group secret) from subshares $f(j) = \sum_{i \in QUAL} f_i(j)$. Though not computed explicitly, the group secret key x is then equal to both $\sum_{i \in QUAL} a_{i0}$ and the Lagrange interpolation involving the shares $\{f(j)\}_{j \in QUAL}$.

1.4 Publicly Verifiable Secret Sharing

While VSS allows verification of the involved shares, the fact of the matter is that such verification can only be done by the involved participants in the secret sharing process. In other words, VSS' verification process is not public. Publicly verifiable secret sharing (PVSS), on the other hand, involves a public verification process, which is useful and desirable in models such as the public bulletin model or blockchain. Here, we delineate Schoenmakers' PVSS scheme and Scrape's optimization of it.

1.4.1 Schoenmakers

A bit different from Shamir's → VSS

1.4.2 Scrape's Optimization

Involves Reed-Solomon codes

1.5 Verifiable Delay Function

1.5.1 Wesolowski's

1.5.2 Pietrzak's

1.5.3 Group Instantiation

RSA group vs class group of $\mathbb{Q}(\sqrt{p})$ (non-trusted setup)

2 Protocols

2.1 PVSS-based

2.1.1 RandHound

One-off protocol

2.1.2 Scrape

Commit-reveal every round, with reconstruction via PVSS

2.1.3 HydRand

Delayed commit-reveal with a leader per round, with reconstruction via PVSS

2.2 DVRF-based

Distributed verifiable random function (DVRF) is quite naturally a distributed version of VRF, where the VRF's secret key is distributed among a group of participants.

Hash of VUF → VRF

Threshold signature can be interpreted under this bucket

2.2.1 RandHerd

Involves threshold Schnorr + CoSi (collective signing = multisig aggregation + communication tree) in a fairly complicated way

2.2.2 Dfinity

League of Entropy's drand

Stalling the network? 100% eventual liveness assumption?

2.2.3 DDH-DVRF

Does not use pairings

2.3 VDF-based

2.3.1 Extending Commit-Reveal

Unicorn: pre-VDF

RANDAO + VDF

2.3.2 Extending Public Randomness

Taking some closing stock price or a block hash + VDF

2.3.3 RandRunner

A VDF chain involving a group of trapdoor VDFs

References

[1] Feldman, Paul. “A practical scheme for non-interactive verifiable secret sharing.” 28th Annual Symposium on Foundations of Computer Science (sfcs 1987). IEEE, 1987.

[2] Dodis, Yevgeniy, and Aleksandr Yampolskiy. “A verifiable random function with short proofs and keys.” International Workshop on Public Key Cryptography. Springer, Berlin, Heidelberg, 2005.