Fully Distributed Verifiable Random Functions and their Application to Decentralised Random Beacons

EuroS&P 2021

David Galindo△†, Jia Liu∗ , Mihai Ordean† , Jin-Mann Wong✤

△Valory, Switzerland

†University of Birmingham, UK

∗Fetch.ai, UK

✤British Antartic Survey, UK

{ david.galindo@valory.xyz, d.galindo@bham.ac.uk }

Verifiable random functions (VRF)

- First introduced by Micali, Rabin and Vadhan in 1999

- A keyed cryptographic hash function
  - ✓ $(pk, sk) \leftarrow KG(\lambda)$
  - ✓ $(v_x, \pi_x) \leftarrow F_{sk}(x)$
  - ✓ Verify(pk, v_x, π_x, x): publicly verifiable
  - ✓ $F_{sk}(x)$ is pseudorandom for PPT adversaries

- IETF is pursuing standardization of a verifiable random function
- Chainlink has a VRF oracle offering

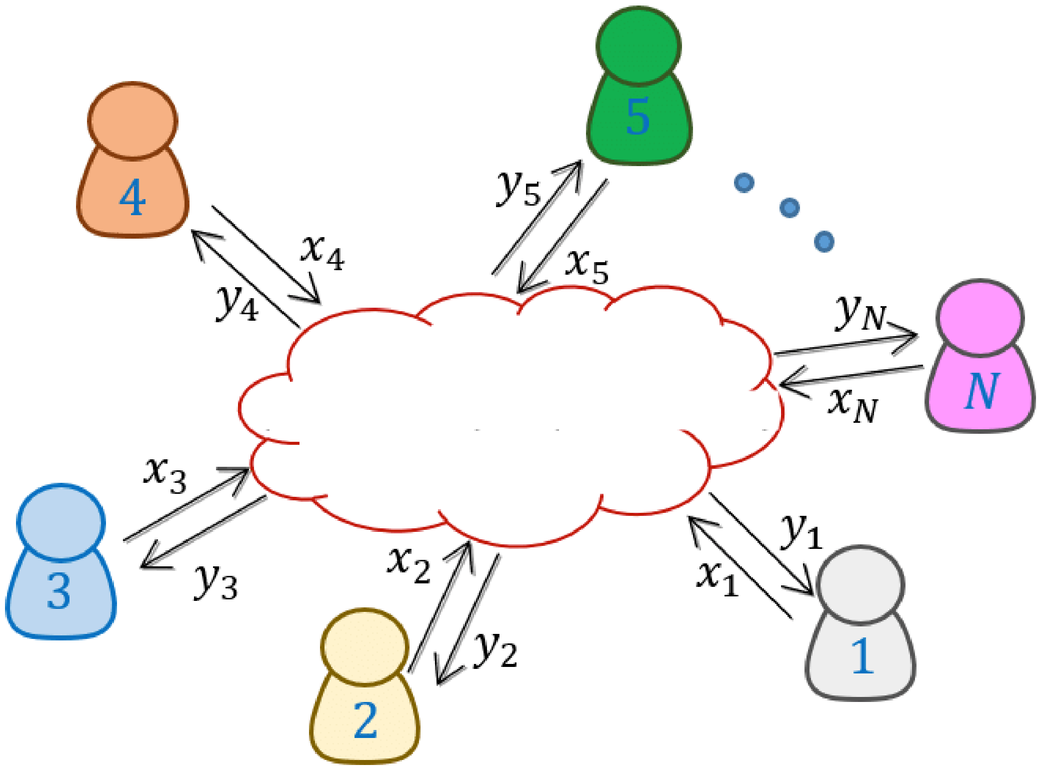
Verifiable (Pseudo-)Randomness: What For?

"Convince those that did not win that the winning party was chosen fairly: pseudorandom, unbiased, unpredictably"

Applications:

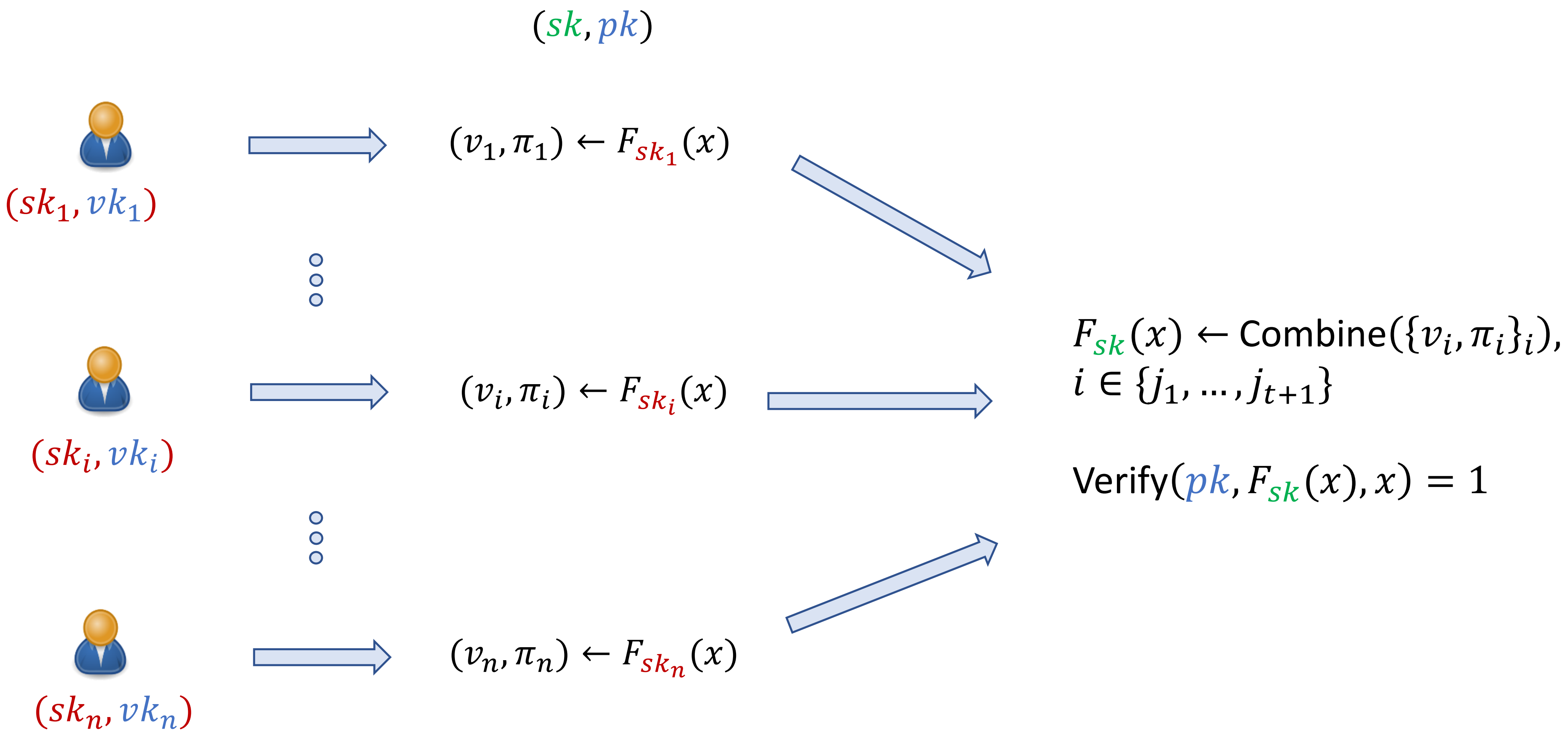
- Lotteries
- Leader selection (Byzantine Agreement, Proof-of-Stake consensus)
- Electronic auctions
- Gaming

DVRFs – Synchronous Setup Phase



Global Public Output: $pk, (vk_1, \ldots, vk_n)$
Local Secret Output: $(sk_1, \ldots, sk_n)$

DVRFs – Asynchronous Randomness Generation



Distributed Random Beacon (DRB)

- Random Beacon: periodical collective randomness sampling
  - Key component for leader selection procedure in consensus protocols, e.g. Tendermint, Ethereum 2.0, OmniLedger, Dfinity and Algorand
- Distributed: avoiding reliance on a central trusted party
  - Robustness
  - Liveness
  - Increased security
  - Asynchronous randomness-generation (non-interactive)
- Distributed computation of an unpredictable and unbiased source of randomness, verifiably

Formalisation of DVRFs

- Admissibility
  - Consistency (correctness), robustness (guaranteed output delivery), uniqueness

- (θ, t, ℓ)-standard pseudo-randomness: no adversary controlling at most $\theta \leq t$ nodes $\{j_1, \ldots, j_\theta\}$ is able to
  - distinguish $F_{sk}(x^\star)$ from random for an adversarial chosen input $x^\star$ on data $\{(v_i, \pi_i) \leftarrow F_{sk_i}(x^\star)\}_i, i \in \{j_1, \ldots, j_\theta\}$

- (θ, t, ℓ)-strong pseudo-randomness: no adversary controlling at most $\theta \leq t$ nodes $\{j_1, \ldots, j_\theta\}$ is able to
  - distinguish $F_{sk}(x^\star)$ from random for an adversarial chosen input $x^\star$ on data $\{(v_i, \pi_i) \leftarrow F_{sk_i}(x^\star)\}_i \cup \{(v_{i'}, \pi_{i'}) \leftarrow F_{sk_i}(x^\star)\}_{i'}, i \in \{j_1, \ldots, j_\theta\}, i' \in \{j_{\theta+1}, \ldots, j_t\}$

Separation results

- **Recap:** (θ, t, ℓ)-standard pseudo-randomness and (θ, t, ℓ)-strong pseudo-randomness: whether the adversary is allowed to obtain any partial randomness evaluation on the challenge plaintext
- **Separation result:** strong pseudo-randomness is strictly stronger than standard pseudo-randomness
- **Real-world separation result**: Algorand is $(0, t, \ell)$-standard pseudorandom but not $(0, t, \ell)$-strong pseudorandom

Construction DDH-DVRF

- DDH-VRF (non-distributed, ESORICS'12)
  - $H(x) \in G$, where G is a DDH group
  - $(sk, pk = g^{sk})$
  - $\left(H(x)^{sk}, \pi_{eqdl}\right)$ where $\pi_{eqdl} = PoK\{(sk): v = H(x)^{sk} \wedge pk = g^{sk}\}$

- DDH-DVRF
  - $(sk, pk = g^{sk})$
  - $\left(H(x)^{sk_1}, \pi_{eqdl}^{1}\right), \ldots, \left(H(x)^{sk_n}, \pi_{eqdl}^{n}\right)$

    $\Rightarrow \; (H(x)^{sk}, \pi)$ where $\pi = \{\pi_{j_1}, \ldots, \pi_{j_{t+1}}\}$
  - Non-compact proof size, strongly pseudorandom under DDH assumption

Construction (pairing-based)

- BLS-VRF [CRYPTO'02]
  - $e : G_1 \times G_2 \rightarrow G_T$
  - $H_1(x) \in G_1$
  - $pk = g_2^{sk}$
  - $\sigma = H_1(x)^{sk}$
  - $e(\sigma, g_2) \stackrel{?}{=} e(H_1(x), pk)$

- Dfinity-DVRF (Threshold BLS)

  $H_1(x)^{sk_1}, \ldots, H_1(x)^{sk_n}$
  $\Rightarrow \left(SHA2(\pi), \pi = H_1(x)^{sk}\right)$

  - Verification keys and public key on G_2
  - Pairing-friendly groups, compact proof
  - Standard pseudorandom under co-CDH assumption

GLOW-DVRF

$$\left(H_1(x)^{sk_1}, \pi_{eqdl}^1\right), \ldots, \left(H_1(x)^{sk_n}, \pi_{eqdl}^n\right)$$
$$\Rightarrow \left(SHA2(\pi), \pi = H_1(x)^{sk}\right)$$

- Verification keys on G_1 and public key on G_2
- Pairing-friendly group, compact proof
- Strongly pseudorandom under the XDH assumption and co-CDH assumption
  - Trick for security reduction: replacing pairing equality check with NIZKs
  - 2.5x faster
- Standard pseudorandom under co-CDH assumption

Benchmarks (I)

https://github.com/fetchai/research-dvrf

Protocol	Curve	Library	Security Level	Proof size (bytes)	Randomness Generation (ms)	Time Ratio	Assumption
GLOW-DVRF	BN256	mcl	100	32	7.38	0.69	co-CDH XDH
	BLS12-381	mcl	128	48	18.67	1.75	
	BN384	mcl	128	48	21.39	2.00	
	BN_P256	RELIC	100	33	33.16	3.10	
DDH-DVRF	Ristretto255	Libsodium	128	1664	10.70	1	DDH
	Curve25519	RELIC	128	1664	65.97	6.17	
Dfinity-DVRF	BN256	mcl	100	32	18.81	1.76	co-CDH
	BLS12-381	mcl	128	48	55.79	5.22	
	BN384	mcl	128	48	60.73	5.68	
	BN_P256	RELIC	100	33	138.36	12.94	

Comparison with existing DRBs

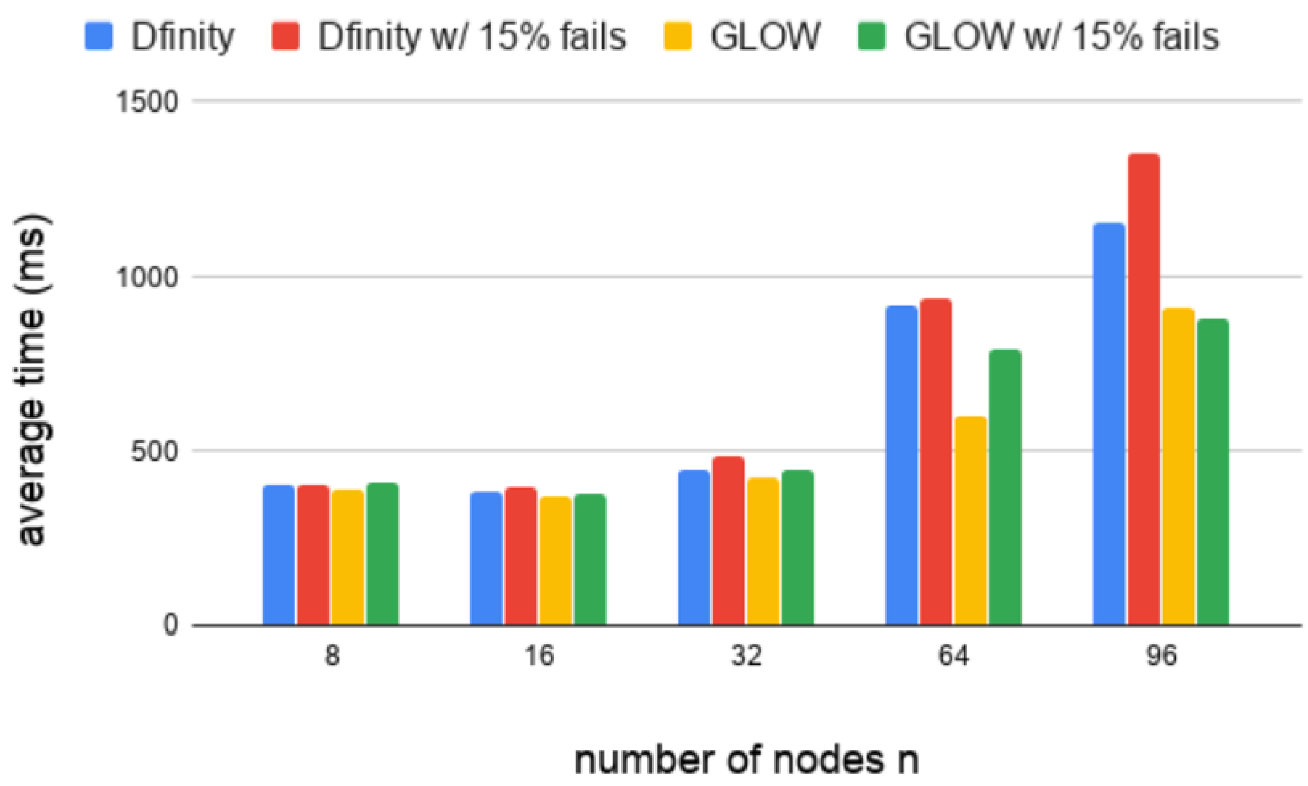
Synchronous distributed setup

Asynchronous distributed randomness computation

Random Beacon Protocol	Standard Pseudorandomness	Strong Pseudorandomness	Strong Bias Resistance	Unpredictability
Algorand-DRB [33]	(–)	✘	✘	✔
Elrond-DRB [27]	(–)	✘	✘	✔
Harmony-DRB [36]	✘	✘	✘	✔
HERB [18]	✔	✔	✘	✔
Orbs-DRB [4]	✔	✔	✘	✔
Ouroboros-Praos-DRB [22]	(–)	✘	✘	✔
Dfinity-DRB [35]	✔	(?)	✔	✔
DDH-DRB [This work]	✔	✔	✔	✔
GLOW-DRB [This work]	✔	✔	✔	✔

Benchmarks (II)

Partially-Synchronous distributed setup
Asynchronous distributed randomness computation
Tendermint consensus nodes are simultaneously DRB nodes



Questions?