

Litepaper

Canary

A ladder of funded discrete-log bounties on Solana:
when a rung falls, markets settle and vaults lock.

v0 · October 2026

Contents

1. Summary

2. The ladder

3. How a rung falls

4. What reads a rung

5. Fees, flows and the CANARY token

6. Architecture

7. Stage

References

A ladder of funded discrete-log bounties on Solana. When a rung falls, rung-event markets settle and post-quantum vaults lock, read straight from the rung's account, with no oracle and no committee.

1. Summary

Quantum risk to elliptic-curve keys now comes with dates attached. Google Quantum AI estimates that 256-bit ECDLP needs fewer than 1,200 logical qubits and 90M Toffoli gates [7]; Project Eleven puts Q-Day possibly as early as 2030 [4]; Allianz counts about 6M BTC, some 30 % of supply, behind exposed public keys [3]. There is still no price for the risk and no signal a program can read. The trigger most discussed, a 256-bit canary, fires at the moment theft becomes possible [1].

Canary funds discrete-log puzzles on curves smaller than secp256k1 and Ed25519 but still beyond classical reach. Each puzzle is a rung. When someone solves one and claims it, a Solana account flips from `Open` to `Solved`. Three things read that account:

- **Bounties** pay the solver in the same instruction.
- **Rung-event markets**, YES and NO tokens fully collateralized in USDC, settle on it.
- **Q-Vaults** close their Ed25519 spending path on it, leaving a Falcon-512 path.

Any other program can read it too.

2. The ladder

There are six rungs in three families. Each rung has two nothing-up-my-sleeve points, G and Q , derived by hash-to-curve from a public seed, so nobody knows the discrete logarithm k with $Q = k \cdot G$.



The ladder, bottom to top by the logical qubits a quantum attack needs (Roetteler et al. 2017 [9]), with each rung's classical Pollard-rho cost.

Rung	Curve	Field / subgroup bits	Classical cost	Logical qubits [9]	Toffoli [9]
Q-160	secp160k1, $y^2 = x^3 + 7$	160 / 161	$\sim 2^{79}$	1,466	3.0e10
Q-192	secp192k1, $y^2 = x^3 + 3$	192 / 192	$\sim 2^{95}$	1,754	5.3e10
Q-224	secp224k1, $y^2 = x^3 + 5$	224 / 225	$\sim 2^{111}$	2,042	8.5e10
C-160	Generic curve over secp256k1's field, ~160-bit subgroup	256 / ~160	$\sim 2^{79}$	~2,330	below the 256-bit count
T-256k	secp256k1, NUMS point	256 / 256	$\sim 2^{127}$	2,330	1.3e11
T-255e	Ed25519, NUMS point	255 / 253	$\sim 2^{126}$	~2,330	~1.3e11

The qubit counts come from Roetteler et al.'s formulas, $9n + 2[\log_2 n] + 10$ qubits and $448n^3 \log_2 n + 4090n^3$ Toffoli gates for an n -bit field [9]. The classical costs use $\sqrt{(\pi n)/12}$ Pollard-rho steps for the $j = 0$ curves ($y^2 = x^3 + b$) and $\sqrt{(\pi n)/4}$ otherwise. The Q-rungs are the SEC 2 curves of those sizes [28]; on every rung both G and Q are nothing-up-my-sleeve points.

Where the ladder starts. The public classical frontier sits around 2^{65} – 2^{68} . Puzzle #130 fell to GPU kangaroo at about $2^{65.5}$ steps, and puzzle #135, about 2^{68} , is still open with a 13.5 BTC bounty [14][15]. Q-160 sits about 2^{11} beyond #135, roughly \$1–10B of rented GPU time. Quantum cost grows roughly linearly with field size, classical cost exponentially, so 160 bits is the lowest rung a GPU farm can't drain while a quantum computer still reaches it first.

Reading a fall. Markets settle on the fact of a fall. The ladder carries its cause:

Observation	Reading
Q-160 falls, C-160 stands	Quantum, or an attack specific to small fields: a strong quantum signal
Q-160 and C-160 both fall	Classical money or a classical algorithm
Q-192 or Q-224 falls	Only a quantum computer or a classical ECDLP breakthrough fits
T-256k or T-255e falls	Theft of secp256k1 or Ed25519 keys is possible

A separate quantum-signal market, which opens with C-160, resolves YES when Q-160 falls while C-160 is still Open.

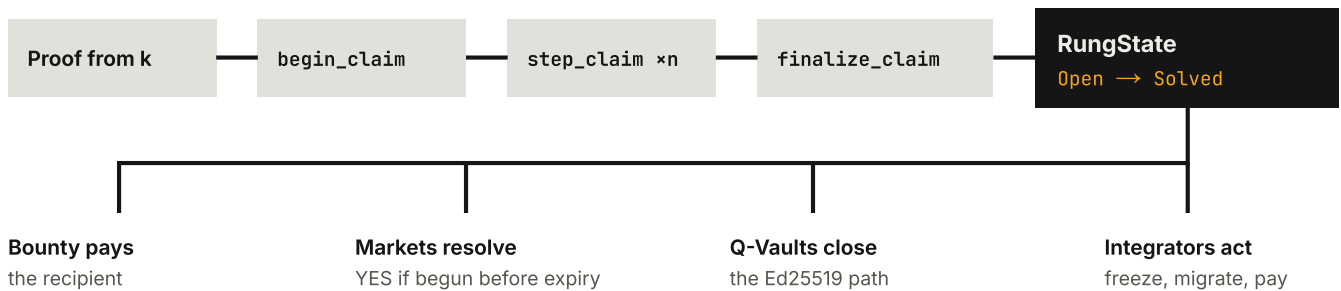
3. How a rung falls

A solver who knows k proves it with a Schnorr proof whose challenge commits to the rung, the recipient and a job nonce, and never reveals k . A plain k in a transaction could be copied by the slot leader or any searcher; a proof bound to the recipient can't be retargeted by anyone who doesn't know k .

$$R = w \cdot G$$

$$e = \text{SHA-256}(\text{"canary.claim.v1"} \parallel \text{program} \parallel \text{rung} \parallel \text{recipient} \parallel \text{nonce} \parallel R \parallel G \parallel Q \parallel \text{transcript}) \bmod n$$

$$s = w + e \cdot k \bmod n \text{ verify: } s \cdot G = R + e \cdot Q$$



One claim, from the proof to every reader of the rung.

The terminal rungs verify in one transaction through Solana's curve syscalls: two `sol_secp256k1_recover` calls perform the scalar multiplications for T-256k (about 345,000 CU in all), and one `sol_curve_multiscalar_mul` for T-255e (about 52,000 CU) [21]. The Q-rungs need field arithmetic Solana has no syscall for, so `step_claim` walks the scalars' bits 16 at a time through a Jacobian accumulator with Shamir's trick, comparing projectively so no inversion is needed: 13 transactions for Q-160, 14 for Q-192 and 17 for Q-224. The rung's `event_ts` is the time `begin_claim` landed, so a fall whose verification spans several transactions still counts before an expiry it beat. A claim must finalize within 300 slots of its start.

4. What reads a rung

Rung-event markets

A market on a rung and an expiry is two SPL tokens. Lock 1 USDC to mint 1 YES and 1 NO, and trade either on any Solana venue. `resolve` is permissionless: YES once the rung reads Solved with `event_ts` on or before the expiry, NO once the expiry passes with the rung Open. Winners redeem 1 USDC per token, less a settlement fee of 1.0 %. Expiries in v0 are 2027-12-31, 2028-12-31 and 2030-12-31 on Q-160, Q-192 and T-256k; the quantum-signal market opens with C-160.

Each rung's open interest is capped at no more than 1 % of the low end of its classical break cost, about \$10M for Q-160, so buying YES and then paying for a brute-force attack never pays.

Across expiries, YES prices draw a market-implied date curve. Between expiries $t_1 < t_2$ with YES prices $p_1 \leq p_2$, the implied conditional chance of a fall is $(p_2 - p_1) \div (1 - p_1)$.

Q-Vault

Every Solana wallet address is an Ed25519 public key; a program-derived address has no private key at all [16][24]. A Q-Vault holds funds at a program address under two keys:

	Ed25519 (the wallet)	Falcon-512 (kept offline)
Before the trigger rung falls	Spends up to an instant budget; larger spends wait out a delay	Spends at any time; vetoes queued Ed25519 spends
After it falls	Rejected permanently	Spends at any time

The instant budget refills only after a full delay passes with no instant spend. A stolen wallet key therefore takes at most one budget before the owner can veto. A Falcon-512 signature (897-byte key, 666-byte signature [16]) covers the exact action, a nonce and the vault's instance salt; it is staged on chain in chunks so every transaction stays within 1,232 bytes [22], and verifies in about 390,000 compute units. Once the trigger reads Solved, `trip`, callable by anyone, records the lockdown for good.

Bounties

Sponsors fund each rung in USDC, irrevocably until a 2040-01-01 sunset. The bounty pays the solver when the rung falls. Each deposit mints Token-2022 NonTransferable receipts one-for-one, the sponsor's share of what the vault holds after the sunset. Sponsorship carries no protocol cut.

Integrators

`RungState` lives at `["rung", id]` under `canary_rungs`. Any program reads it in any instruction by checking its owner, discriminator, address and version. A status feed serves the six states as JSON with the slot they reflect.

5. Fees, flows and the CANARY token

Line	Rate
Settlement fee	1.0 % of winning redemptions
Minting and merging pairs	None
Sponsorship	0 % protocol cut
Q-Vault	Free

Fees are swept 70 % to the buyback vault and 30 % to the bounty vaults, split across rungs by open interest weighted by the slots it was held. A position opened and closed in the same slot carries no weight. Rungs people hedge against grow their own bounties.

The buyback vault is a USDC account owned by the markets program. Its USDC leaves only in one transaction that swaps it for CANARY, Canary's token, through Jupiter and burns every CANARY bought; if the burn can't happen, nothing moves:

CANARY

Supply	1,000,000,000, fixed
Launch	100 % to one public launch: no presale, team, investor, airdrop or reserved allocation, no vesting
Burn	The protocol's 70 % of every settlement fee buys CANARY and burns it

burned per settlement = CANARY bought with 1 % × winning redemptions × 70 %

6. Architecture

Program	Holds	Upgrade policy
canary_rungs	Rungs, bounty vaults, claim verification	Immutable after audit
canary_markets	Markets, resolution, fees	3-of-5 multisig, 7-day timelock in v0; frozen in v1
canary_vault	Q-Vaults	3-of-5 multisig, 7-day timelock in v0; later a Falcon-gated Squads multisig

Around the programs:

- **Parameters:** a deterministic parameter generator with a published transcript.
- **Off-chain services:** an indexer, a crank that resolves markets, sweeps fees and runs the buyback, and a public API with the status feed.
- **Clients:** browser clients that prepare every transaction for the user's own wallet, and an offline Falcon-512 SDK.

Every instruction fits a legacy 1,232-byte transaction, so every wallet can sign it.

Why Solana. A transaction gets 1.4M compute units and verification can be staged across several; both terminal rungs are nearly free to verify with native syscalls [21]. Falcon-512 verification already runs in SBF [17][18]. Addresses being Ed25519 keys gives Q-Vault native demand, and long-dated positions are cheap to hold.

CANARY stays outside the trigger. The trigger must be credibly neutral, so CANARY is never collateral, a resolution input, a vote on rungs or vault rules, or a payout backstop; the rungs program becomes immutable once audited. Collateral is USDC because a fall would most likely move crypto prices when payouts come due.

7. Stage

Canary v0 consists of:

- the three programs, with Q-160, Q-192, Q-224, T-256k and T-255e in the parameter suite and C-160 joining when its point count is certified;
- the parameter generator and its transcript;
- the web app for sponsors, traders, vault users and solvers;
- the claim CLI;
- the indexer, the crank, and the public API with the status feed;
- CANARY, launched in full to the public, with the protocol's fee share buying and burning it.

The full design is in the [whitepaper](#).

References

The numbers follow the whitepaper's source list: [1] BitMEX Research, "Avoiding An Unnecessary Quantum Freeze" (2026) · [3] Allianz Research, quantum finance special (2026-07-29) · [4] The Block, Project Eleven Q-Day report (2026-05-06) · [7] Babbush et al., arXiv:2603.28846 · [9] Roetteler, Naehrig, Svore, Lauter, arXiv:1706.06752 · [14] Bitcoin.com News, puzzle status (2026-06-23) · [15] Bitcoin-Puzzle-Info · [16] Jump Crypto, "Quantum migration paths for Solana" (2026-04-27) · [17] Solana Compass, Squads Falcon-512 test and Blueshift Vector (2026-10-13) · [18] SIMD-0461 Falcon syscall PR · [21] Agave execution budget constants · [22] Solana docs, Transactions · [24] Solana docs, Accounts.