PoSW from Skip List

February 27, 2018

Abstract

1 Introduction

1.1 PoSW

1. PoSW vs PoW vs Time-lock puzzles
2. Original construction from depth-robust graphs

1.2 Related Work

1. Time-release crypto [RSW00, BGJ$^+$15, May93, MMV11]
2. Original construction [MMV13] using depth-robust graphs [MMV13, EGS75]
3. [CP18] construction doesn't require depth-robustness

1.3 Our Contribution

1. Intuitive construction of POSW based on skip lists [Pug90]
2. Larger gap in proof generation and verification using PRPs and sloth hash function [LW17]
3. PRP used instead of a hash function and the input to it is squared before

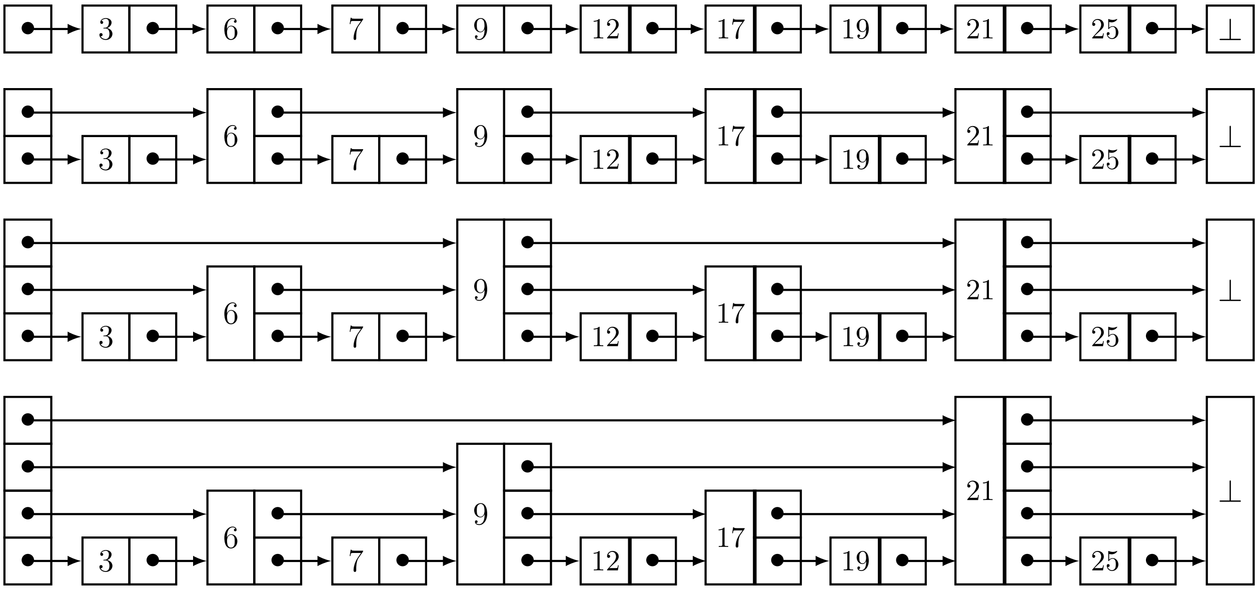


Figure 1: Linked list vs. skip lists

2 Preliminaries

2.1 Formal Definitions

1. PoSW

2.

2.2 Skip list

1. [Pug90]

2. randomised vs. deterministic (binary)

3. Figure

2.3 [CP18] Construction

2.4 The Sloth Hash Function

1. [LW17]

2. Assumptions: computing square-root requires logarithmically many squarings

3 Warm-up: PoSW from Skip Lists

We start with basic construction that uses an ensemble of PRPs and the skip list, and then show that it is a proof of sequential work.

3.1 Construction

The construction takes as input a time parameter $N = 2^n$ for $n \in \mathbb{N}$ and two statistical parameters $w, t \in \mathbb{N}$. We assume an ensemble of random permutations $P := P_0, \ldots, P_n$ with $P_i : \{0,1\}^{(i+1)\cdot w} \to \{0,1\}^{(i+1)\cdot w}$ sampled uniformly at random from $\mathcal{P}_{(i+1)\cdot w}$, the set of permutations on $(i+1)\cdot w$-bit-long strings in $\{0,1\}^*$. Let P^{-1} denote the "inverse" oracle of P.

1. The verifier sends the statement $\chi \in \{0,1\}^{(n+1)\cdot w}$ to the prover
2. The prover computes the sequence of states $\sigma_0, \ldots, \sigma_N$ and sets $\phi = \sigma_N$ and $\phi_p = \sigma_0, \ldots, \sigma_N$. (We will see later how one can trade-off space for time just like in [CP18].)
3. The verifier, on receipt of ϕ, challengers the prover on t random leaf nodes $\gamma_1, \ldots, \gamma_t$, where $\gamma_i \in [N]$.

Definition 1. An adversary A, with oracle access to P and P^{-1}, on an input $x \in \{0,1\}^w$ outputs a P-sequence $x_0, \ldots, x_s$ of length s if

1. $x \subset x_0$, where $\subset$ denotes that x is a continuous substring of x_0 (i.e., x_0 is of the form $a \circ x \circ b$ for some $a, b \in \{0,1\}^*$).
2. For all $j \in [0, s-1]$, there exists some $i \in [0, n]$ such that $P_i(x_j) \subset x_{j+1}$, where $\subset$ denotes that some continuous substring of $P_i(x_j)$ of length w is present as a continuous subsequence of x_{j+1}.

Claim 0.1. *The probability that an adversary outputs a P-sequence of length s making (strictly) less than s sequential queries is*

$$2q \cdot \frac{Q + \sum_{i=0}^{s} |x_i^2|}{2^{(t-1)w}},$$

where q denotes the total number of queries that the adversary is allowed to make to the random permutations and Q their total length.

"Proof". The three ways that A can output a P-sequence $x_0, \ldots, x_s$ making less than s queries are given below.

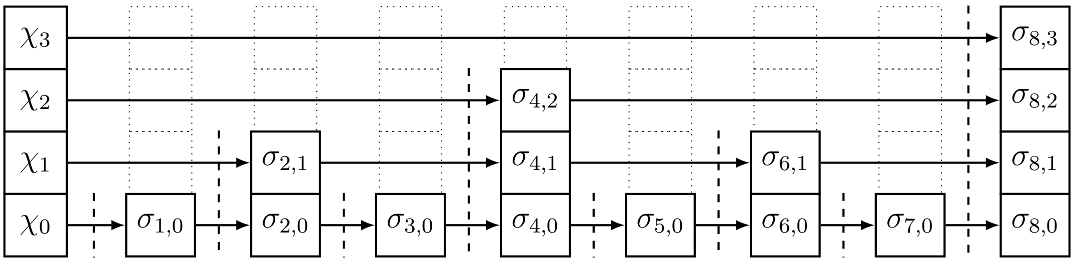


Figure 2: Schematic diagram for $N = 8$. The statement is of the form $\chi = \chi_0 \circ \cdots \circ \chi_3$ and is the initial state σ_0 in the construction. The proof, ϕ, is the final state $\sigma_8 = \sigma_0 \circ \cdots \circ \sigma_3$. The vertical lines represent an application of the permutation P with the length indicative of the exact permutation: e.g., the rightmost vertical line denotes an application of P_4 to the input $\chi_3 \circ \sigma_{4,2} \circ \sigma_{6,1} \circ \sigma_{7,0}$.

1. Lucky guess of a value of P: for some $i \in [s], j \in [n]$ it holds that $P(x_i) \subset x_{i+1}$ and the adversary *did not* make the query $P(x_i)$. As P is random, the probability of this event can be upper-bounded by
$$q \cdot \frac{\sum_{i=0}^{s} |x_i^2|}{2^{(t-1)w}}?$$
2. Collision: The x_js were not computed sequentially. That is it holds that for some $0 \leq j < k \leq s-1$ a query x_j is made in round j and a query x_k is made in round k where $P(x_j) \subset x_k$. Again, since P is uniformly random, the probability is
$$q \cdot \frac{Q^2}{2^{(t-1)w}}?$$
3. 1. or 2. occurs with P^{-1}: similar bounds hold.

The original bound follows by a union bound. □

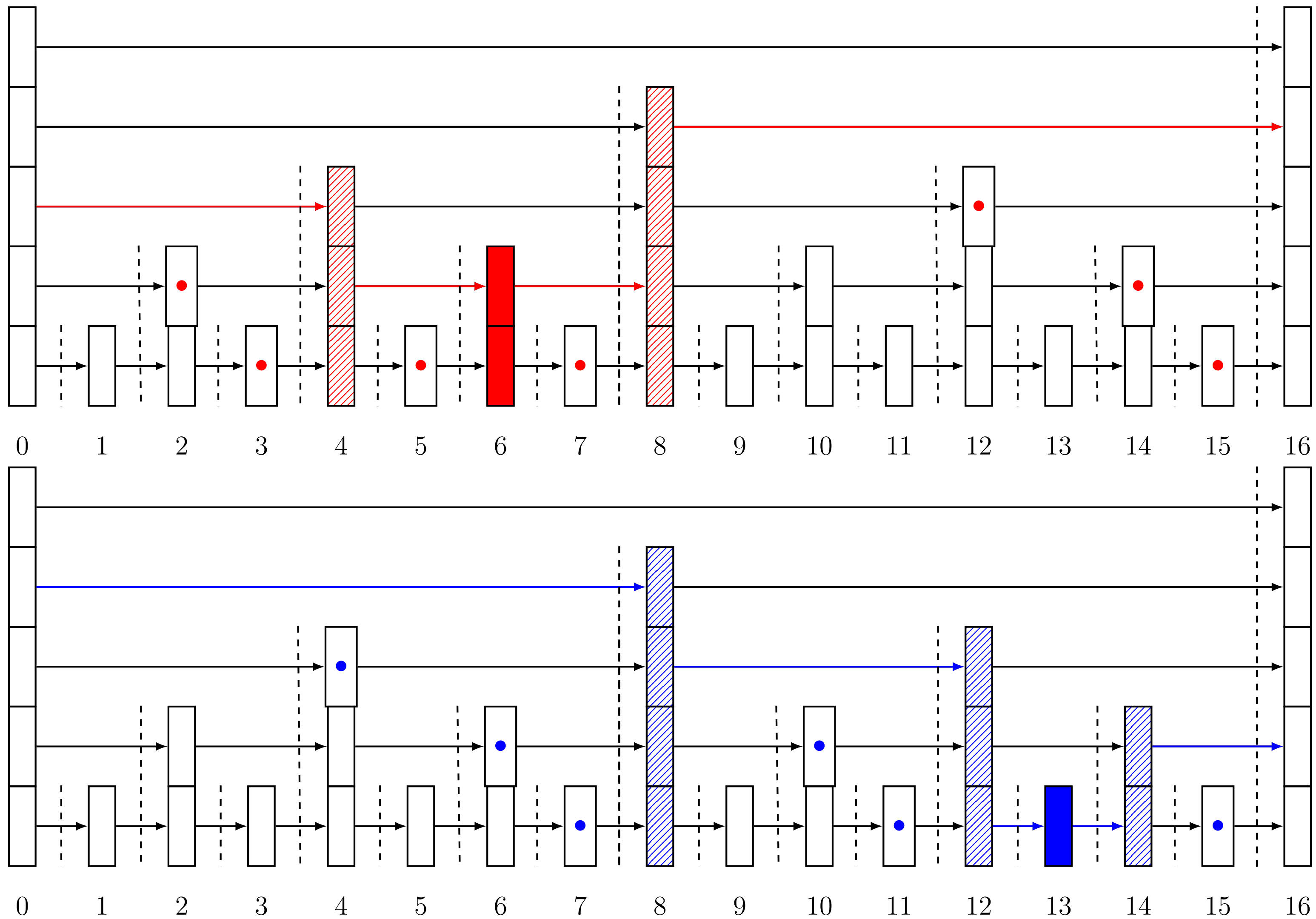
3.2 Trade-off

4 Main Construction

References

[BGJ+15] Nir Bitansky, Shafi Goldwasser, Abhishek Jain, Omer Paneth, Vinod Vaikuntanathan, and Brent Waters. Time-lock puzzles

Figure 3: The red and blue box represent two (independent) challenges $c_1 = 6$ and $c_2 = 13$. The red (resp., blue) path is the shortest path from 0 to 16 that goes through 6 (resp., 13). The (sub)states with red (resp., blue) bullets are required for the verification of the red (resp., blue) path. The shaded boxes are recomputed during the verification.



from randomized encodings. Cryptology ePrint Archive, Report 2015/514, 2015. http://eprint.iacr.org/. (Cited on page 1.)

[CP18] Bram Cohen and Krzysztof Pietrzak. Simple proofs of sequential work. Cryptology ePrint Archive, Report 2018/183, 2018. https://eprint.iacr.org/2018/183. (Cited on pages 1, 2 and 3.)

[EGS75] P. Erdös, R.L. Graham, and E. Szemerédi. On sparse graphs with dense long paths. *Computers and Mathematics with Applications*, 1(3):365 – 369, 1975. (Cited on page 1.)

[LW17] Arjen K. Lenstra and Benjamin Wesolowski. Trustworthy public randomness with sloth, unicorn, and trx. *IJACT*, 3(4):330–343, 2017. (Cited on pages 1 and 2.)

[May93] Timothy C. May. Timed-release crypto. *http://www.hks.net/cpunks/ cpunks-0/1460.html*, 1993. (Cited on page 1.)

[MMV11] Mohammad Mahmoody, Tal Moran, and Salil Vadhan. Time-lock puzzles in the random oracle model. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 39–50. Springer Berlin Heidelberg, 2011. (Cited on page 1.)

[MMV13] Mohammad Mahmoody, Tal Moran, and Salil Vadhan. Publicly verifiable proofs of sequential work. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, pages 373–388, New York, NY, USA, 2013. ACM. (Cited on page 1.)

[Pug90] William Pugh. Skip lists: A probabilistic alternative to balanced trees. *Commun. ACM*, 33(6):668–676, June 1990. (Cited on pages 1 and 2.)

[RSW00] Ronald L. Rivest, Adi Shamir, and David Wagner. Time-lock puzzles and timed-release crypto. *Technical Report MIT/LCS/TR-684, MIT*, February 2000. (Cited on page 1.)