A note on an ECDLP-based PoW model

Alessio Meneghetti[1], Massimiliano Sala[2], and Daniele Taufer[3]

[1] University of Trento, Trento, Italy
alessio.meneghetti@unitn.it
[2] University of Trento, Trento, Italy
maxsalacodes@gmail.com
[3] University of Trento, Trento, Italy
daniele.taufer@unitn.it

Abstract

We lay the foundations for a blockchain scheme, whose consensus is reached via a proof-of-work algorithm based on the solution of consecutive discrete logarithm problems over the point group of elliptic curves. In the considered architecture, the curves are pseudorandomly determined by block creators, chosen to be cryptographically secure and changed at every epoch. Given the current state of the chain and a prescribed set of transactions, the curve selection is fully rigid, therefore trust is needed neither in miners nor in the scheme proposers.

1 Introduction

A *proof-of-work* (PoW) is a procedure that allows a prover to demonstrate that it is very likely to having performed a specific amount of computational work within a prescribed interval of time.

This concept has been formalized in 1999 [19], although previous instances of delaying functions conceived for similar purposes had appeared earlier [2, 9, 12, 14, 17, 20, 30].

Since 2008, PoW-methods [24] have been attracting a considerable interest as Bitcoin [27] introduced a PoW-based consensus algorithm, which puts miners in competition for solving a cryptographic challenge. Bitcoin's consensus relies on a hashcash system [3, 4], whose workload may be easily adjusted with a fastly verifiable output. Despite their high efficiency and easy implementation, all the hashcash-based protocols share a common limitation: the huge amount of computations employed by nodes becomes useless after the consensus is reached. This aspect has been raising environmental concerns and many solutions have been proposed to reduce these energy-intensive computer calculations.

A promising countermeasure to this issue is the adoption of *bread pudding protocols* [19]. They face the aforementioned problem by perfoming a computational work that is reusable either for practical [11, 13, 26, 33], cryptographical [19, 29] or mathematical [36] reasons. Moreover, the latter class of systems encloses several protocols that are meant to be research propellants [5], namely designed to boost the commitment upon the solution of difficult mathematical problems.

Along the same line, we propose a blockchain architecture with a PoW-consensus algorithm based on the solution of the *Discrete Logarithm Problem* over the point groups of elliptic curves (ECDLP). This approach does not aim at reducing the energy consumption, but at adding a scientific scope to the PoW mechanics, meanwhile achieving supplementary security features.

The idea of basing the PoW on ECDLP has already appeared in other works [18, 21], as this problem is widely studied and applied in cryptographic protocols. However, the considered curves does not usually fulfil the standard security criteria [6], especially for what concerns the *fully rigidity*: the network has to initially trust an authority that is providing the curve parameters.

In this work we radically solve this issue by designing a PoW-system based on elliptic curves that are changing over the time. Since the curves are pseudo-randomly constructed and satisfy general security conditions, a malicious user could attack the chain only by breaking the ECDLP for an immense class of elliptic curves, which is currently considered infeasible.

An extended version of this work may be found in [23].

This paper is organized as follows: after a quick summary of the ECDLP in Section 2, we deline the proposed blockchain architecture in Section 3 and its blocks construction in Section 4. The strong points of this system are discussed in Section 5, while in Section 6 future work directions are suggested.

2 ECDLP

The ECDLP is a renown problem that consists of finding an integer $N \in \mathbb{N}$ such that the N-th multiple of a *base point* P of an elliptic curve E over a finite field equals another given point Q, i.e. $Q = N \cdot P$.

Solving ECDLP for a well-chosen curve E is considered to be a difficult challenge. Currently the best known *general* attacks are Baby-Step Giant-Step [32] and Pollard's Rho - Kangaroo algorithms [28], which have an asymptotic complexity of $O(\sqrt{|E|})$, where $|E|$ is the size of E. The introduction of Semaev's polynomials [31] have suggested the existence of subexponential algorithms to solve ECDLP, however no clear evidence has emerged. Pairings-based attacks [15, 25], Index calculus [1, 22, 35] and Xedni calculus [34] have been recently being studied, but none of them seem to significantly reduce the problem complexity so far.

3 A sample blockchain architecture

To show how our PoW works, we introduce a schematic sample ledger architecture, but our algorithm may easily be adapted for any blockchain scheme. Our architecture is based on two types of blocks:

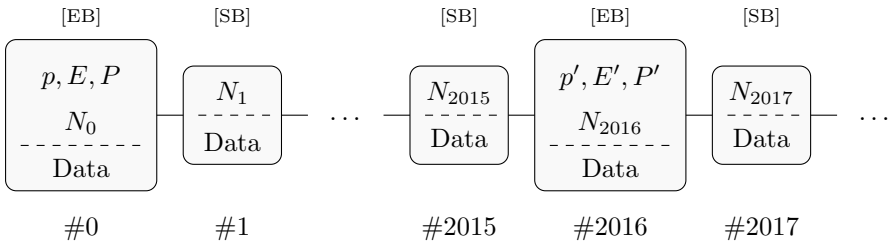
[EB] An *Epoch Block* contains, aside from the header and a list of transactions, a prime number p, an elliptic curve E defined over $\mathbb{F}_p$ and a base point P of E, all to be determined by the proposing miner.

Moreover, it encloses as PoW an integer $N \in \{0, \dots, |E| - 1\}$ to be discovered by the proposing miner such that $N \cdot P$ is the point of E that is deterministically determined from the header of the block.

These EBs occur once every 2016 blocks in the blockchain.

[SB] The *Standard Blocks* are just a light version of the EB blocks, they are constructed in the same way except for p, E, P, which are inherited from the last EB block of the chain.

SBs constitute the vast majority of the blocks of the chain.



EBs basically define the setting (curves and base points) on which the discrete logarithm PoWs will have to be solved in the following epoch. They are slightly heavier to be produced and verified but occur rarely (roughly once every two weeks with a BTC-like difficulty adjustment).

4 Block construction

In this section we define the envisioned specifications of blocks, whose motivation will be analyzed in Section 5.

First, we need a deterministic function `P_Gen` to construct a point on a given elliptic curve E from a prescribed hash digest h, which we treat as an integer for simplicity. The following is a concrete example of such a function.

```
function P_Gen(h,E)
  i = 0
  while #{points of E with x-coord = h + i} = 0:
    i = i + 1
  P = (h + i, *) point of E with 0 ≤ * < p/2
  return P
```

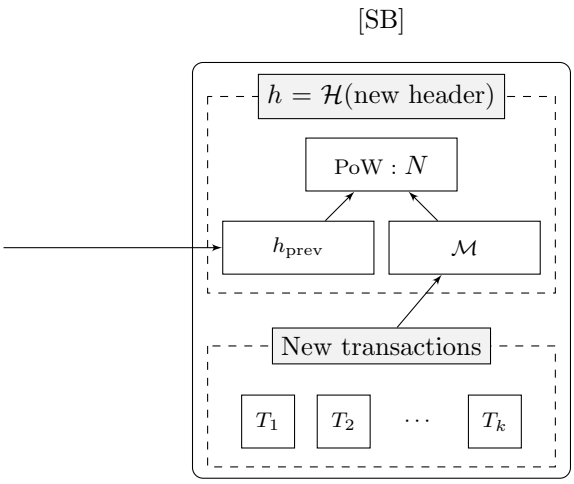
We notice that the points determined by the above function are affine by construction. The hash $\mathcal{H}$ that we propose to use in the following is SHA3-512 [7], which provides a satisfying collision resistance even against post-quantum attacks, but one might conceivably replace it with another properly constructed one.

Standard Blocks

A minimal model of a SB consists of a list of valid transactions and a header, which comprises their Merkel root $\mathcal{M}$, the hash of the previous header h_{prev} and an integer N solving

$$\text{PoW}: \quad \texttt{P\_Gen}(\mathcal{H}(h_{\text{prev}}||\mathcal{M}), E) = N \cdot P,$$

where E and P are the ones defined in the last EB.



Epoch Blocks

An EB is a thick version of a SB, namely it is constructed in a similar fashion but it encloses three additional data: the prime p, the elliptic curve E over $\mathbb{F}_p$ and the base point P of E.

- Generating p

  The prime number p is the responsible of the expected running time of the PoW. Its size is determined by the difficulty parameter d, whose tuning depends on the block production ratio that a designer wants to obtain. Therefore we do not discuss the choice of d but we refer to the BTC implementation [8] or to more structured models such as personalized difficulty adjustments [10]. Our goal is to produce a prime number of the prescribed size and satisfying known properties of secutity (for a detailed description, see the full paper [23]).

  Given the difficulty parameter d and the hash of the previous header h, we propose the generation of such a prime number p as follows.

```
function p_Gen(d, h)
  repeat
    h = H(h)
    p = NextPrime(h mod 2^2d)
  until p satisfies the required properties
  return p
```

- Generating E

  We aim at generating pseudorandom elliptic curves for which no attacks are currently known, i.e. satisfying a number of security properties [6, 23].

  Let h be the previous block header, we suggest to generate the curve as follows.

```
function E_Gen(p, h)
  i = 0
  repeat
    i = i + 1
    A_E = H(h + i)
```

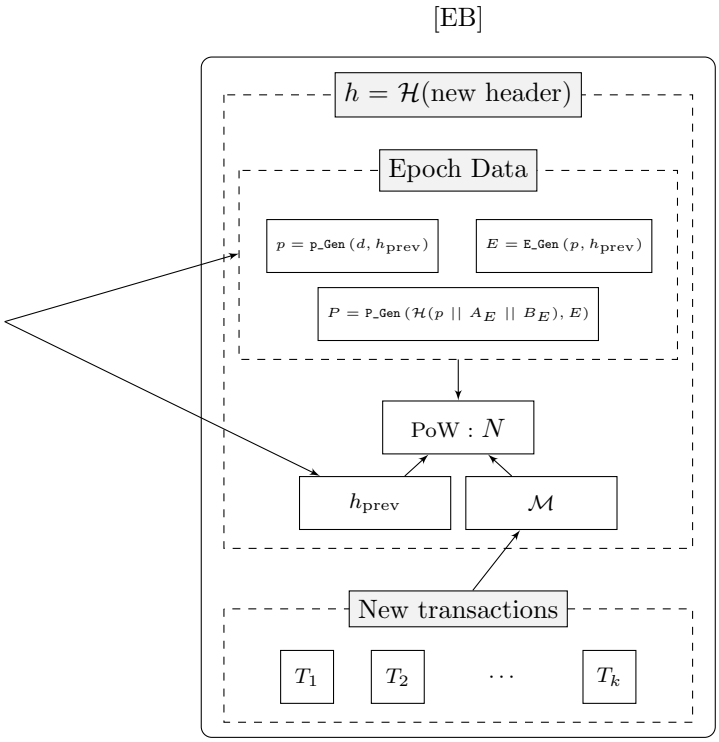
```
  B_E = H(A_E)
  E defined by y^2 = x^3 + A_E x + B_E over F_p
until E is an EC satisfying security properties
return E
```

- Generating P

  The base point we prescribe for an EB and its subsequent epoch is

$$P = \texttt{P\_Gen}(\mathcal{H}(p \mid\mid A_E \mid\mid B_E), E).$$

The new epoch parameters are manufactured before the PoW production, which therefore depends on them.



Despite the verification of SBs is extremely fast, EBs are slower to be checked since verifiers need to test that all the curve parameters involved have been properly constructed, running several types of mathematical algorithms such as primality testing, finite fields operations and points counting.

5 Method discussion

Here we discuss motivation and advantages of the presented choices.

First, this PoW model involves many different mathematical algorithms of wide interest, for which this blockchain may represent a concrete research propellant. Furthermore, it might also provide a public collection of cryptographically secure elliptic curves of moderate size.

Apart from its scientific usefulness, it conveys many desirable security properties. The challenges involved do not rely on a given curve of questionable provenance but on the *generic* difficulty of the ECDLP, which is much more fair to be trusted. Thus, we find it aims at embracing the decentralization ideals that lead to cryptocurrencies creation: even the mathematical objects involved are publicly manufactured, no trust is required even in the authors or the proposing entities.

As for blocks forgery, we point out that both SBs and EBs comprise a PoW which depends on the entire block, together with the previous one. This means that any counterfeit in any position of the chain results into an incorrect final block, which may be easily detected from the network.

Moreover, it is hard to conceive shortcuts for the PoW production: for a given difficulty parameter d we expect a d-bits secutity of the *general* ECDPL by using $p \approx 2^{2d}$, unless attacks outperforming Pollard's rho are discovered. Furthermore, common base field operations speed ups are avoided by making use of non-exceptional primes, ensuring a fair and general problem to be solved equally for every miner. In fact, neither specific algorithms nor dedicated hardware may be used for solving such a general problem, of which easy cases are carefully avoided [23].

Besides security, the curves we propose are *fully rigid* as defined in [6]: their construction is entirely explained in terms of the previous block, which cannot be controlled by a malicious actor since there is no room for miner choices (such as nonces). Even assuming that the transactions of the previous block might be chosen *ad hoc*, an attacker who wants to impose a particular curve during the next epoch has to brute-force invert the hash $\mathcal{H}$ at the cost of one ECDLP solution for each attempt, until a desired hash digest is obtained, within the time needed for the entire network to solve a single ECDLP. We consider this scenario unachievable under realistic assumptions. However, even assuming that a miner succeeds in breaking or finding a shortcut for solving the ECDLP over the current epoch curve, it would benefit from the chance of consistently winning the PoW-challenge only until the next epoch. In any case, this miner would get no advantage in the choice of the next epoch curve. We find this eventuality a fair reward for such an unlikely and scientifically unexpected achievement.

As regards the difference between EBs and SBs, we point out that the bulk of miner's work consists of the ECDLP solution: we expect good parameters to be generated in EBs in a time which is linear in the difficulty parameter [16] whereas the asymptotic difficulty of ECDLP solution is exponential in it.

Since the curves creation appears not to be computationally demanding when compared to the actual PoW, then lazy miners do not have any substantial advantage in skipping it.

6 Conclusion

We have proposed a new PoW-based blockchain model based on *general* ECDLP, highlighting the desirable properties that such a scheme provides in terms of scientific relevance, security and pure decentralization ideals.

The past proposals [18, 21] have the high merit of introducing ECDLP as a problem whose solution provides consensus, but we felt compelled to remove the suspiscious choice of the curve serving as a common battlefield for miners. The novel multi-curve approach introduced in the current work wipes out this grey area by providing a fully transparent scheme.

It may be interesting to produce an actual implementation of the proposed scheme, obtaining practical time measurments and efficiency considerations. A subsequent engaging project might address the resistance of such a protocol to the known attacks under real-world assumptions, comparing the obtained results with outcomes of existing cryptocurrencies.

Finally, different types of PoW might be conceived in a similar fashion, possibly employing problems which are thought to resist even to quantum attacks.

Aknowledgments

The results presented here have been carried on within the EU-ESF activities, call PON Ricerca e Innovazione 2014-2020, project Distributed Ledgers for Secure Open Communities. We thank the Quadrans Foundation for its support.

References

[1] A. Amadori, F. Pintore, M. Sala, *On the discrete logarithm problem for prime-field elliptic curves*, Finite Fields and Their Applications, Vol. 51, pp. 168–182 (2018), URL: https://doi.org/10.1016/j.ffa.2018.01.009.

[2] S. Ar, J. Cai, *Benchmarks Using Numerical Instability*, SODA (1994), URL: https://dl.acm.org/citation.cfm?id=314476.

[3] A. Back, *Hashcash*, (1997), URL: http://www.cypherspace.org/hashcash.

[4] A. Back, *Hashcash - A Denial of Service Counter-Measure*, (2002), URL: http://www.hashcash.org/papers/hashcash.pdf.

[5] M. Ball, A. Rosen, M. Sabin, P. N. Vasudevan, *Proofs of Useful Work*, IACR (2017), URL: https://eprint.iacr.org/2017/203.pdf.

[6] D. J. Bernstein, T. L. Lange, *Safecurves: choosing safe curves for elliptic-curve cryptography*, URL: https://safecurves.cr.yp.to/.

[7] G. Bertoni, J. Daemen, M. Peeters, G. van Assche, R. van Keer, *Keccak implementation overview*, (2012), URL: https://keccak.team/files/Keccak-implementation-3.2.pdf.

[8] Bitcoin team, *PoW implementation*, (2018), URL: https://github.com/bitcoin/bitcoin/blob/master/src/pow.cpp.

[9] J. Cai, R. J. Lipton, R. Sedgewick, A. C. Yao, *Towards uncheatable benchmarks*, IEEE (1993), URL: https://doi.org/10.1109/SCT.1993.336546.

[10] C. Chou, Y. Lin, R. Chen, H. Chang, I. Tu, S. Liao, *Personalized Difficulty Adjustment for Countering the Double-Spending Attack in Proof-of-Work Consensus Protocols*, arXiv (2018), URL: https://arxiv.org/abs/1807.02933.

[11] CureCoin Team, *2019 Curecoin Model (White Paper draft)*, (2019), URL: https://curecoin.net/white-paper.

[12] C. Dwork, M. Naor, *Pricing via Processing or Combatting Junk Mail*, Annual International Cryptology Conference (1992), URL: https://doi.org/10.1007/3-540-48071-4_10.

[13] H. Finney, *RPOW - Reusable Proofs of Work*, (2004), URL: https://nakamotoinstitute.org/finney/rpow/index.html.

[14] M. K. Franklin, D. Malkhi, *Auditable Metering with Lightweight Security*, Financial Cryptography (1997), URL: https://doi.org/10.1007/3-540-63594-7_75.

[15] G. Frey, H. Rück, *A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves*, Mathematics of Computation (1994), URL: https://www.jstor.org/stable/2153546?seq=1.

[16] S. D. Galbraith, J. Mckee, *The Probability That The Number Of Points On An Elliptic Curve Over A Finite Field Is Prime*, IEEE (2000), URL: https://www.math.auckland.ac.nz/~sgal018/cm.pdf.

[17] D. M. Goldschlag, S. G. Stubblebine, *Publicly Verifiable Lotteries: Applications of Delaying Functions*, Financial Cryptography (1994), URL: http://dl.acm.org/citation.cfm?id=647502.728319.

[18] M. Hastings, N. Heninger, E. Wustrow, *The Proof is in the Pudding: Proofs of Work for Solving Discrete Logarithms*, IACR (2018), URL: https://eprint.iacr.org/2018/939.pdf.

[19] M. Jakobsson, A. Juels, *Proofs of Work and Bread Pudding Protocols (Extended Abstract)*, Secure Information Networks (1999), URL: https://doi.org/10.1007/978-0-387-35568-9_18.

[20] A. Juels, J. Brainard, *Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks*, NDSS (1999), URL: https://www.ndss-symposium.org/ndss1999/cryptographic-defense-against-connection-depletion-attacks.

[21] M. Lochter, *Blockchain as cryptanalytic tool*, IACR (2018), URL: https://eprint.iacr.org/2018/893.pdf.

[22] G. McGuire, D. Mueller, *A New Index Calculus Algorithm for the Elliptic Curve Discrete Logarithm Problem and Summation Polynomial Evaluation*, IACR (2017), URL: https://eprint.iacr.org/2017/1262.pdf.

[23] A. Meneghetti, M. Sala, D. Taufer, *A new ECDLP-based PoW model*, ArXiv (2019), URL: https://arxiv.org/abs/1911.11287.

[24] A. Meneghetti, M. Sala, D. Taufer, *A survey on PoW-based consensus*, AETiC, Vol. 4, No. 1 (2020). URL: http://aetic.theiaer.org/archive/v4/v4n1/p2.pdf.

[25] A. J. Menezes, T. Okamoto, S. A. Vanstone, *Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field*, IEEE (1993), URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=259647.

[26] A. Miller, A. Juels, E. Shi, J. Katz, *Permacoin: Repurposing Bitcoin Work for Data Preservation*, IEEE (2014), URL: https://www.microsoft.com/en-us/research/publication/permacoin-repurposing-bitcoin-work-for-data-preservation.

[27] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, (2008), URL: https://bitcoin.org/bitcoin.pdf.

[28] J. M. Pollard, *Monte Carlo Methods for Index Computation (mod p)*, Mathematics of Computation (1978), URL: http://doi.org/10.2307/2006496.

[29] R. L. Rivest, A. Shamir, *PayWord and MicroMint: Two simple micropayment schemes*, International Workshop on Security Protocols (1997), URL: http://doi.org/10.1007/3-540-62494-5_6.

[30] R. L. Rivest, A. Shamir, D. A. Wagner, *Time-lock Puzzles and Timed-release Crypto*, Massachusetts Institute of Technology Cambridge (1996), URL: https://dl.acm.org/citation.cfm?id=888615.

[31] I. A. Semaev, *Summation polynomials and the discrete logarithm problem on elliptic curves*, IACR (2004), URL: https://eprint.iacr.org/2004/031.pdf

[32] D. Shanks, *Class Number, a Theory of Factorization and Genera*, Proceedings of Symposium of Pure Mathematics, Vol. 20, pp. 415–440 (1969).

[33] A. Shoker, *Sustainable Blockchain through Proof of eXercise*, IEEE (2017), URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8171383.

[34] J. H. Silverman, *The Xedni Calculus and the Elliptic Curve Discrete Logarithm Problem*, Designs, Codes and Cryptography, Vol. 20, pp. 5–40 (2000), URL: https://doi.org/10.1023/A:1008319518035.

[35] J. H. Silverman, J. Suzuki, *Elliptic Curve Discrete Logarithms and the Index Calculus*, Advances in Cryptology – ASIACRYPT ' 98, pp. 120–125 (1998) URL: https://doi.org/10.1007/3-540-49649-1_10.

[36] K. Sunny, *Primecoin: Cryptocurrency with Prime Number Proof-of-Work*, (2013), URL: http://primecoin.io/bin/primecoin-paper.pdf.