A new ECDLP-based PoW model

Alessio Meneghetti, Massimiliano Sala, Daniele Taufer

June 2, 2020

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 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 he is very likely to having performed a specific amount of computational work within a prescribed interval of time [34].

This concept has been formalized in 1999 [28], although previous instances of delaying functions conceived for similar purposes had appeared earlier [3, 12, 18, 20, 25, 29, 42].

Since 2008, PoW-methods have been attracting a considerable interest as Bitcoin [38] introduced a PoW-based consensus algorithm, which puts miners in competition for solving a cryptographic challenge. Bitcoin’s consensus relies on a hashcash system [4, 5], 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 [28]. They face the aforementioned problem by performing a computational work that is reusable either for practical [17, 19, 37, 47], cryptographical [28, 41] or mathematical [52] reasons. Moreover, the latter class of systems encloses several protocols that are meant to be research propellants [6], namely designed to boost the commitment upon the solution of difficult mathematical problems.

Along the same line, we have proposed [33] 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). In this work, we provide that germinal proposal with precise mathematical foundations and further implementation details.
The idea of basing the PoW on ECDLP has already appeared in other works\cite{27,30}, as this problem is widely studied and applied in cryptographic protocols. However, the considered curves does not usually fulfil the standard security criteria\cite{7}, 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.

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 3.1 and 3.2. The strong points of this system are discussed in Section 4, including a theorem on the security of our system, while in Section 5 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. $Q = N \cdot P$.

Here we are only interested in elliptic curves over prime fields $\mathbb{F}_p$ and determined by their short Weierstrass equation $y^2 = x^3 + Ax + B$. Solving ECDLP for a curve $E$ over large fields is considered to be a difficult challenge except for degenerate cases.

2.1 The general case

Currently the best known general attacks are Baby-Step Giant-Step\cite{46} and Pollard’s Rho - Kangaroo algorithms\cite{40}, which have an asymptotic complexity of $O(\sqrt{|E|})$, where $|E|$ is the size of $E$. These are general parallel collision-finding algorithms, which work over any groups, i.e. no properties of the underlying structure but the operation definition are used.

The introduction of Semaev’s polynomials\cite{45} have suggested the existence of subexponential algorithms to solve ECDLP, however no clear evidence has emerged. Pairings-based attacks\cite{22,36}, Index calculus\cite{2,32,49} and Xedni calculus\cite{48} have been recently being studied, but none of them seem to significantly reduce the problem complexity of the general case, so far.

2.2 Special cases

There are some families of curves whose ECDLP is known to be easier than the general case, namely there are algorithm for efficiently solving it. Consequently, these curves have to be carefully avoided for designing a ECDLP-based protocol. The following is a concise summary of those particular attacks, the curve on which they may be efficiently applied and how we avoid them.
| Attack                  | It applies on curves               | To avoid it: use                | Ref.   |
|------------------------|-----------------------------------|---------------------------------|--------|
| Weil-descent           | over composite fields             | prime fields                    | [21]   |
| Polig-Hellman          | of composite orders               | prime orders                    | [24]   |
| Semaev, Satoh-Araki, Smart | anomalous                        | non-anomalous curves           | [44]   |
| Menezes-Okamoto-Vanstone | low embedding degree             | high embedding degree           | [39]   |
| Frey-Rück              | low embedding degree              | high embedding degree           | [43]   |
| Wiener-Zuccherato      | low CM discriminant               | high CM discriminant            | [50]   |

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 two integer $N_1$ and $N_2 \in \{0, \ldots, |E| - 1\}$ to be discovered by the proposing miner such that $N_i \cdot P$ are points of $E$ 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).

In order to give the specifications of our blocks we need a deterministic function $P_{\text{Gen}}$ 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.

```python
function P_{\text{Gen}}(h,E)
    i = 0
    while \#\{points of E with x-coord = h + i\} = 0:
        i = i + 1
```
\begin{align*}
i & = i + 1 \\
P & = (h + i, *) \text{ point of } E \text{ with } 0 \leq * < p/2 \\
\text{return } P
\end{align*}

We notice that the points determined by the above function are affine by construction. The hash \( H \) that we propose to use in the following is SHA3-512 [8], which provides a satisfying collision resistance even against post-quantum attacks, but one might conceivably replace it with another properly constructed one.

We also assume that all proposing miners use prescribed signature algorithms and we denote with \( \sigma_k(m) \) the signature of the string \( m \) obtained by the miner with signing key \( k \).

### 3.1 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 \( h_{\text{prev}} \) and a pair of integers \((N_1, N_2)\) solving

\[
\text{PoW : } \begin{cases} 
P_{\text{Gen}}(H(h_{\text{prev}}), E) = N_1 \cdot P, \\
P_{\text{Gen}}(H(\mathcal{M}), E) = N_2 \cdot P.
\end{cases}
\]

where \( E \) and \( P \) are defined in the last EB.

### 3.2 Epoch Blocks

An EB is a thick version of a SB, namely it is constructed in a similar fashion but it encodes 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 run 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 [9] or to more structured models.
such as personalized difficulty adjustments [14]. Our goal is to produce a prime number of the prescribed size and satisfying the following properties.

**EXCEPTIONALITY PROPERTIES**

1. \( p \) is not a Crandall prime [16], i.e. not of the form \( 2^k - c \) for a relatively small and positive integer \( c \).

2. \( p \) is neither a Generalized Mersenne prime [51] nor a More Generalized Mersenne prime [15], i.e. it may not be written as \( p(m) \) for some integer \( m \) and polynomial \( p \) with very small coefficients and number of monomials.

3. \( p \) is not Montgomery-friendly [1, 26, 10], i.e. it may not be obtained as \( 2^n(2^\beta - \gamma) - 1 \) for small positive integers \( \alpha, \beta, \gamma \).

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.

```python
function p_Gen (d, h)
    repeat
        h = \( H(h) \)
        p = NextPrime(h mod 2^{2d})
    until p satisfies exceptionality properties
    return p
```

• Generating \( E \)

We aim at generating pseudorandom elliptic curves for which no efficient attacks are currently known, i.e. satisfying the following properties.

**SECURITY PROPERTIES**

1. The number of points of \( E \) is prime and different from \( p \).

2. The embedding degree \( B \) is greater than 20, i.e. \( |E| \nmid p^B - 1 \) for every \( 1 \leq B \leq 20 \).

3. Let \( D \) be the CM field discriminant, defined as

\[
D = \begin{cases} 
\Delta & \text{if } \Delta \equiv 1 \text{ mod } 4, \\
4\Delta & \text{otherwise,}
\end{cases} \quad \Delta = \text{SquareFreePart}(t^2 - 4p),
\]

where \( t \) is the trace of \( E \). Then we require \( D > 2^{40} \).

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

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

5
\[ B_E = \mathcal{H}(A_E) \]

\[
E \text{ defined by } y^2 = x^3 + A_EX + B_E \text{ over } \mathbb{F}_p
\]

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

\[ P = P_{\text{Gen}}(\mathcal{H}(p \ || \ A_E \ || \ B_E), E). \]

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

\[ [\text{EB}] \]

\[ h = \mathcal{H}(\text{new header}) \]

Epoch Data

\[ p = p_{\text{Gen}}(d, h_{\text{prev}}) \quad E = E_{\text{Gen}}(p, h_{\text{prev}}) \]

\[ P = P_{\text{Gen}}(\mathcal{H}(p \ || \ A_E \ || \ B_E), E) \]

PoW : \((N_1, N_2)\)

\[ h_{\text{prev}} \quad M \]

New transactions

\[ T_1 \quad T_2 \quad \ldots \quad T_k \]

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.

4 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 provides 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.

The existence of different types of block in blockchains has become common, as it is considered suitable for tackling the problem of scalability [35].

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 security of the general ECDPL by using \(p \approx 2^{2d}\), unless attacks outperforming Pollard’s rho are discovered. Moreover, common base field operations speed ups are avoided by making use of not-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. Also, the constructed curves fulfil the known security criteria [7]:

- working over prime fields avoids Weil-descent attacks;
- searching for curves of prime order prevents from Polig-Hellman attacks;
- since \(p \neq |E|\) the curves are not anomalous so Smart, Semaev, Satoh-Araki attacks do not apply;
- the embedding degree we suggest is greater than 20 as required by SEC1 [13], which prevents pairing attacks such as Menezes-Okamoto-Vanstone (based on Weil Pairing) and Frey-Rück (based on Tate-Lichtenbaum Pairing);
- attacks to curves with low CM discriminant are prevented by requiring it higher than \(2^{40}\), as for Brainpool Standard Curves [31].

**Theorem 4.1.** Let us assume that the current epoch is endowed with the curve \(E\) and its base-point \(P\). Let \(\sigma\) be a deterministic digital signature algorithm and \(\mathcal{M}\) be a proposing miner with fixed signing key \(k\). If \(\mathcal{M}\) exhibits a valid block, then it has solved at least one generic instance of ECDLP on \(E\).

**Proof.** By definition of our PoW, the given block is valid if and only if it contains \((N_1, N_2)\) such that

\[
\begin{align*}
Q_1 &= P_{\text{Gen}}(\mathcal{H}(\sigma_k(h_{\text{prev}})), E) = N_1 \cdot P, \\
Q_2 &= P_{\text{Gen}}(\mathcal{H}(\mathcal{M}), E) = N_2 \cdot P.
\end{align*}
\]
Since $E$ and $P$ are determined by the epoch and $h_{\text{prev}}$ is determined by the previous block, the proposing miner has no control on them. Moreover, $\sigma$ is deterministic and $k$ is fixed, so $Q_1 = \mathsf{P}_\mathsf{Gen}(H(\sigma_k(h_{\text{prev}})), E)$ cannot be influenced by the miner. Therefore the miner must solve $Q_1 = N_1 \cdot P$.

\[\square\]

**Remark 4.2.** Even if in the previous proof we do not consider the equation $Q_2 = N_2 \cdot P$, we believe that it adds extra security. Indeed, it is unlikely that a miner can avoid solving $Q_2 = N_2 \cdot P$ unless $\mathcal{M}$ has computed a multiple of $P$ with any $m$, $Q = mP$, and solved the hash preimage equation $H(\mathcal{M}) = x$, where $x$ is the $x$-coordinate of $Q$.

Besides security, the curves we propose are fully rigid as defined in [7]: 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 $H$ 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.

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 [23] 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.
5 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 [27, 30] have the high merit of introducing ECDLP as a problem whose solution provides consensus, but we felt compelled to remove the suspicious choice of the curve serving as a common battlefield for miners.

It may be interesting to produce an actual implementation of the proposed scheme, obtaining practical time measurements 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. Further studies may also be carried on other types of curve models, such as Edwards or Montgomery curves. Even though this is likely to improve the overall performance of this scheme, it should be observed that it contrasts with our declared intention of making use of general objects.

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

Acknowledgments

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] T. Acar, D. Shumow, Modular Reduction without Pre-Computation for Special Moduli, Microsoft Research (2016), URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/modmul_no_precomp.pdf

[2] 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

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

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

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

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

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

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

[10] J. W. Bos, C. Costello, H. Hisil, K. Lauter, *Fast Cryptography in Genus 2*, IACR (2012), URL: https://eprint.iacr.org/2012/670.pdf

[11] W. Bosma, J. Cannon, C. Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput., Vol. 24, pp. 235–265 (1997), URL: http://magma.maths.usyd.edu.au/magma

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

[13] Certicom Research, *SEC 1: Elliptic Curve Cryptography*, (2000), URL: https://www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/1_01sec1.pdf

[14] 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

[15] J. Chung, A. Hasan, *More Generalized Mersenne Numbers*, (2003), URL: http://cacr.uwaterloo.ca/techreports/2003/corr2003-17.ps

[16] R. E. Crandall, *Method and apparatus for public key exchange in a cryptographic system*, Google Patents (1992), URL: https://patents.google.com/patent/US5159632A/en

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

[18] 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

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

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

[21] G. Frey, H. Gangl, *How to disguise an elliptic curve (Weil descent)*, Talk at ECC 98 (1998), URL: http://www.cacr.math.ualberta.ca/conferences/1998/ecc98/frey.ps

[22] G. Frey, H. Röck, *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

[23] 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

[24] P. Gaudry, F. Hess, N. P. Smart, *Constructive and destructive facets of Weil descent on elliptic curves*, Journal of Cryptology (2002), URL: https://doi.org/10.1007/s00145-001-0011-x

[25] 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

[26] M. Hamburg, *Fast and compact elliptic-curve cryptography*, IACR (2012), URL: https://eprint.iacr.org/2012/309.pdf

[27] 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
[28] M. Jakobsson, A. Juels, *Proofs of Work and Bread Pudding Protocols (Extended Abstract)*, Secure Information Networks (1999), URL: https://doi.org/10.1007/978-0-387-35668-9_18

[29] 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

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

[31] M. Lochter, J. Merkle, *Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation*, (2010), URL: https://tools.ietf.org/html/rfc5639

[32] 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

[33] A. Meneghetti, M. Sala, D. Tauer, *A Note on an ECDLP-based PoW Model*, CEUR Proceedings - DLT, 2020.

[34] A. Meneghetti, M. Sala, D. Tauer, *A survey on PoW-based consensus*, AETiC, Vol. 4, No. 1, 2020.

[35] A. Meneghetti, M. Sala, D. Tauer, *A survey on efficient parallelization of blockchain-based smart contracts*, AETiC, Vol. 3, No. 5, 2019.

[36] 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

[37] 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

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

[39] S. Pohlig, M. Hellman, *An Improved Algorithm for Computing Logarithms over GF(p) and its Cryptographic Significance*, IEEE (1978), URL: https://ee.stanford.edu/~hellman/publications/28.pdf

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

[41] 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

[42] 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

[43] T. Satoh, K. Araki, *Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves*, Comm. Math. Univ. Sancti Pauli, Vol. 47, pp. 81–92 (1998).

[44] I. A. Semaev, *Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p*, Mathematics of Computation (1998), URL: http://doi.org/10.1090/S0025-5718-98-00887-4
[45] I. A. Semaev, *Summation polynomials and the discrete logarithm problem on elliptic curves*, IACR (2004), URL: https://eprint.iacr.org/2004/031.pdf

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

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

[48] 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

[49] 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

[50] N. Smart, *The discrete logarithm on elliptic curves of trace one*, Journal of Mathematical Cryptology (1999), URL: http://doi.org/10.1007/s001459900052

[51] J. A. Solinas, *Generalized Mersenne Numbers*, (1999), URL: http://www.cacr.math.uwaterloo.ca/techreports/1999/corr99-39.pdf

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

[53] M. J. Wiener, R. J. Zuccherato, *Faster attacks on elliptic curve cryptosystems*, SAC (1998), URL: https://link.springer.com/chapter/10.1007/3-540-48892-8_15