Pinocchio Coin: Building Zerocoin from a Succinct Pairing-based Proof System

George Danezis
Microsoft Research
Cambridge, UK

Cédric Fournet
Microsoft Research
Cambridge, UK

Markulf Kohlweiss
Microsoft Research
Cambridge, UK

Bryan Parno
Microsoft Research
Redmond, USA

ABSTRACT

Bitcoin is the first widely adopted distributed e-cash system and Zerocoin is a recent proposal to extend Bitcoin with anonymous transactions.

The original Zerocoin protocol relies heavily on the Strong RSA assumption and double-discrete logarithm proofs, long-standing techniques with known performance restrictions. We show a variant of the Zerocoin protocol using instead elliptic curves and bilinear pairings. The proof system makes use of modern techniques based on quadratic arithmetic programs resulting in smaller proofs and quicker verification. We remark on several extensions to Zerocoin that are enabled by the general-purpose nature of these techniques.

Categories and Subject Descriptors

K.4.4 [**Computers and Society**]: Electronic Commerce-Payment schemes, Security

Keywords

Zero-knowledge Proofs; anonymous electronic cash; bitcoin; zerocoin.

1. INTRODUCTION

The central component of Bitcoin is a public log or ledger of transactions. Each transaction entry in the log associates a bitcoin amount with a public key. A new entry is either created by contributing to the authenticity of the log by checking and hashing previous transactions and performing proofs of work; or by using the private key corresponding to an existing entry to sign a new entry. The latter transfers the bitcoin amount of the existing entry to the owner of the public key of the new entry. As regards privacy, the log publicly links coins to their successive owner's keys.

PETShop'13, November 4, 2013, Berlin, Germany.

ACM 978-1-4503-2489-2/13/11 ...$15.00.
http://dx.doi.org/10.1145/2517872.2517878 .

Zerocoin [MGGR13] is an anonymous decentralized e-cash system that uses Bitcoin both as an append-only bulletin board and a backing currency. Zerocoin uses a fixed bitcoin amount, i.e. all zerocoins have the same denomination. Instead of a public key, coins are identified by a commitment C to a pair of fresh, random secrets: a serial number s and an opening r, kept by the owner of the coin.

To guarantee anonymity, a zerocoin spend transaction involves revealing s and proving knowledge of r for any C in a large, public collection of previously-logged commitments $C_0, \ldots, C_{n-1}$. The opening to the commitment of the coin being spent is never revealed but is used to compute a proof π for a signature of knowledge that replaces the conventional signature of a bitcoin spend transaction. The signature of knowledge proves that the spending party can open one of the commitments to the serial number, i.e. that (1) she knows a $C \in (C_1, \ldots, C_n)$ and (2) that $C = g^s h^r$ (the commitment scheme is a Pedersen commitment). By hiding which commitment can be opened in this way, Zerocoin provides anonymity. At the same time, the uniqueness of the serial number prevents double spending.

[MGGR13] uses an Strong RSA based accumulator to prove $C \in (C_1, \ldots, C_n)$, thus all commitments C_i must be prime numbers from an interval $[A, A^2)$, for some fixed integer A, to guarantee that the product of two commitments is outside this interval. These constraints can be met, but Strong RSA based constructions like this can be quite brittle and it would be desirable to have an alternative construction based on prime-order groups. Another complication arises from the proof that $C = g^s h^r$ being about a value C that is already secret and an exponent for the group in which the accumulator is defined. Thus it is what is usually referred to as a double-discrete logarithm proof.

We address both of these issues by making use of Pinocchio [PHGR13], a novel pairing-based proof system with a very efficient implementation

Pinocchio can prove languages of the form $L = \{(c_k)_{k\in[m']} \mid \exists\, (c_k)_{k\in[m'..m-1]} : c_0 = 1 \wedge (\mathbf{V}\cdot\mathbf{c}) \circ (\mathbf{W}\cdot\mathbf{c}) - (\mathbf{Y}\cdot\mathbf{c}) = 0\}$, where $\mathbf{V}$, $\mathbf{W}$, $\mathbf{Y}$ are $d \times m$ matrices over a field $\mathbb{F}_p$ for integers $d, m', m,\ m' \leq m$.[1] $P = (\mathbf{V}, \mathbf{W}, \mathbf{Y})$ is called a quadratic

[1]We write $[n]$ for the set $\{0, \ldots, n-1\}$. We write $\mathbf{X}\cdot\mathbf{y}$ for the multiplication of a matrix with a vector $\mathbf{z} = (\sum_{k\in[n]} X_{ik} y_k)_{i\in[d]}$ and $\mathbf{x}\circ\mathbf{y}$ for the pointwise (Hadamard) product $\mathbf{z} = (x_i y_i)_{i\in[d]}$.

arithmetic program (QAP) over field $\mathbb{F}$ of degree d and size m and the problem of deciding whether P can accept a subvector $(c_0, \ldots, c_{m'-1})$ with $c_0 = 1$ was shown by [GGPR13] to be **NP** complete.

In particular the language L allows us to encode arbitrary input output relations for an arithmetic circuit with d multiplication gates. Intuitively, c encodes wire values, and each row in $\mathbf{V}$ and $\mathbf{W}$ represents a linear combination of wires that will be the left and the right input of a multiplication gate respectively.

Our construction of Zerocoin uses two simple insights: First, $C \in (C_0, \ldots, C_{n-1})$ can be represented by checking that the arithmetic circuit $\prod_i (C - C_i) = 0$. Second, instead of proving knowledge of r, we can prove knowledge of $h_0, \ldots, h_{\nu-1}$ for a security parameter ν of the commitment scheme such that, for $j \in [\nu]$, $(h_j - 1)(h_j - h^{(2^j)}) = 0$ and $C = S \prod_j h_j$, where $S = g^s$ can be publicly computed. Instead of requiring C to be a prime in $[A, A^2)$, the commitment can now be defined over any field in which the discrete logarithm problem is hard.

We are left with one remaining difficulty. If we use the efficient pairing groups of Pinocchio, computing discrete logarithms in the exponent field $\mathbb{F}_p$ with $p \approx 256$ is easy. We could switch to non-standard and larger pairing groups, but this seems undesirable as it would bring down the overall performance of the proof system. Instead we propose to compute C in an extension field $\mathbb{F}_{p^\mu}$ of size $p^\mu > 2048$.

We do not claim that our construction is always desirable over the existing Strong RSA construction. One drawback of our scheme is that the trusted setup instead of being a single RSA modulus N is now the evaluation key of a Pinocchio QAP—a more complex object. It is also unclear whether ultimately a proof of arithmetic circuits in extension fields will scale better than a double discrete logarithm proof. One performance characteristic that is, however, drastically improved is the size of the proof π which no longer depends linearly on ν. Another more qualitative advantage is the availability of an alternative construction based on a different number theoretic problem.

2. CONSTRUCTION

In presenting our protocol we assume limited familiarity with Zerocoin [MGGR13] and Pinocchio [PHGR13].

- $\mathsf{Setup}(1^\kappa)$. On input a security parameter, select or generate a pairing-friendly elliptic curve setup $\mathcal{G}$ for curves of order p to be used by Pinocchio.

  Select random generators $g, h \in \mathbb{F}_{p^\mu}$ such that $\langle g \rangle = \langle h \rangle$ is a large multiplicative subgroup of $\mathbb{F}_{p^\mu}$ of order $q | p^\mu - 1 \approx 2^\nu$.

  Run evaluation key generation $EK_P \leftarrow \mathsf{KeyGen}(P, \mathcal{G})$ for the publicly-verifiable zero-knowledge variant of Pinocchio for verifying **NP** relations expressed as arithmetic constraints. P is a QAP over $\mathbb{F}_p$ of degree and size $O((n + \kappa)\mu^2)$ for the following witness relation, where all operations and values are over $\mathbb{F}_{p^\mu}$:

  $\big((C_0, \ldots, C_{n-1}, S), (h_j)_{j=1}^\kappa\big) \in R_L \Leftrightarrow$
  $\forall j (h_j - 1)(h_j - h^{(2^j)}) = 0 \wedge \prod_i (S \prod_j h_j - C_i) = 0.$

  Output $params = (\mathcal{G}, p, q, g, h, EK_P)$ as the Zerocoin parameters.

- $\mathsf{Mint}(params)$. Select a serial number and opening $s, r \in \mathbb{F}_q \setminus 1$ and compute $C = g^s h^r$ in $\mathbb{F}_{p^\mu}$. Set $skc = (s, r)$ and output (C, skc).

- $\mathsf{Spend}(params, C, skc, C_0, \ldots, C_{n-1})$. If $C \notin (C_i)_{i=0}^{n-1}$ output $\perp$. Compute $S = g^s$, and $h_j = h^{2^j r_j}$, for $j \in [\kappa]$, where the $r_j \in \{0, 1\}$ are such that $r = \prod 2^j r_j$. Then run the Pinocchio prove algorithm $\pi \leftarrow \mathsf{Compute}(EK_P, (C_0, \ldots, C_{n-1}, S, (h^{2^j})_{j=0}^{\nu-1}), (h_j)_{j=1}^\kappa)$ and output (π, s).

- $\mathsf{Verify}(params, \pi, s, C_0, \ldots, C_{n-1})$. Check that $\mathsf{Verify}(EK_P, (C_0, \ldots, C_{n-1}, g^s, (h^{2^j})_{j=0}^{\nu-1}), \pi) = 1$.

3. PERFORMANCE

Recall that $\mathbb{F}_{p^\mu}$ is the Galois field extension of $\mathbb{F}_p$ (that is, $[p]$), defined as the quotient $\mathbb{F}_p[x]/P(x)$ of the polynomials in x with coefficients in $\mathbb{F}_p$ divided by $P(x) = x^\mu - \omega$, for some fixed $\omega \in \mathbb{F}_p$ such that $P(x)$ is irreducible.

We represents elements $A \in \mathbb{F}_{p^\mu}$ by the coefficients $(a_i)_{i \in [\mu]}$ such that $A(x) = \sum_i a_i x^i$. Addition is just word-wise addition: $(a_i)_{i \in [\mu]} + (b_i)_{i \in [\mu]} = (a_i + b_i)_{i \in [\mu]}$. Multiplication is a linear combination of μ^2 word multiplications:

$$(a_i)_{i \in [\mu]} * (b_j)_{j \in [\mu]} = \Big(\sum_{i+j=k} (a_i * b_j) + \sum_{i+j=k+\mu} (\omega * a_i * b_j) \Big)_{k \in [\mu]}.$$

We use $\mathbb{F}_{p^\mu}$ for Pedersen commitments, with exponents in $\mathbb{F}_q$. Fast exponentiation consists of $\nu - 1$ extended multiplications, where $h^r = \prod_{i \in [\nu]} h^{(2^i r_i)}$ and $r = \sum 2^i r_i$ Hence, computing h^r and proving that each of the h_i is either 1 or $h^{(2^i)}$ takes $\mu^2(2\nu - 1)$ word multiplications.

Where Pinocchio really shines in the size of its proof and the cost of proof verification. Contrary to the almost prohibitive proof size of Strong RSA zerocoins of 50kB, the proof size of 344 bytes for Pinocchio zerocoins is comparable with existing bitcoin transactions.

4. DISCUSSION

This is only a very preliminary case study and we do not have a full implementation or security analysis yet. There is also one feature of the Zerocoin protocol that is not covered by our construction. The original Zerocoin construction allows to sign a transaction string R by using the Fiat-Shamir based proof system in signature of knowledge [CL06] mode. On the upside, the analysis of our protocol does no longer rely on Random Oracles. Moreover, we are aware of three ways to extend our protocol: (i) compute s as the hash of a public key and use the corresponding secret key to sign R; (ii) construct a signature of knowledge by using the techniques of [Har11] to turn make the proof simulation extractable; (iii) perform part of the proof using a Fiat-Shamir based proof system and fall back on the Random Oracle model to obtain signatures of knowledge.

We are excited about the potential of using a general-purpose verifiable computation protocol like Pinocchio for custom protocol design. Pinocchio already allows to compile arithmetic circuits from C-like programs.

For instance, this make it very easy to replace our commitment scheme $C = g^s h^r$, by another commitment scheme like $C = \mathsf{HMAC}(r, s)$, e.g. based on SHA-256. One could also

imagine, more complex spend protocols that involve multiple commitments or commitments with a balance controlled by a scripting language akin to *Bitcoin script*.

5. REFERENCES

[CL06] Melissa Chase and Anna Lysyanskaya. On signatures of knowledge. In *CRYPTO*, 2006.

[GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In *EUROCRYPT*, 2013.

[Har11] Kristiyan Haralambiev. *Efficient cryptographic primitives for non-interactive zero-knowledge proofs and applications*. PhD thesis, 2011.

[MGGR13] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *IEEE Symposium on Security and Privacy*, 2013.

[PHGR13] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *IEEE Symposium on Security and Privacy*, 2013.