Faster probabilistic verification of one or more Pinocchio proof

January 16, 2017

The Pinocchio verifier uses pairing operations. The verifier requires 12 pairings. However, using the fact that the pairing is bilinear, and that some of the pairing inputs repeat themselves, we can reduce the number of required operations, assuming we are willing to use *probabilistic* verification, where there is a $1/|\mathbb{F}_r|$ probability of accepting a false proof.

The basic claim used for the probabilistic verification is

Claim 0.1. *Suppose G is a group of prime order r. (The G we are thinking of here is the multiplicative subgroup of order r in $\mathbb{F}_{p^k}$ which is the pairing target group). Suppose that $a_1, \ldots, a_t \in G$ are not all equal to one. Chooose random elements $r_1, \ldots, r_t \in \mathbb{F}_r$, then the probability that*

$$a_1^{r_1} \cdots a_t^{r_t} = 1$$

is at most $1/r$.

More significant savings can be made when we want to "batch" proofs - meaning that we want to check whether there is at least one bad proof in a sequence of proofs (all using the same verification key). Details follow.

1 optimizing one Pinocchio verification

We want to check

1. $e(\pi_A, vk_A) = e(\pi'_A, g_2)$
2. $e(vk_B, \pi_B) = e(\pi'_B, g_2)$
3. $e(\pi_C, vk_C) = e(\pi'_C, g_2)$
4. $e(\pi_K, vk_\gamma) = e(vk_x + \pi_A + \pi_C, vk^2_{\beta\gamma})e(vk^1_{\beta\gamma}, \pi_B)$.
5. $e(vk_x + \pi_A, \pi_B) = e(\pi_H, vk_Z) \cdot e(\pi_C, g_2)$

Note first that the above checks are equivalent to:

1. $e(\pi_A, vk_A) \cdot e(\pi'_A, -g_2) = 1$
2. $e(vk_B, \pi_B) \cdot e(\pi'_B, -g_2) = 1$
3. $e(\pi_C, vk_C) \cdot e(\pi'_C, -g_2) = 1$

4. $e(\pi_K, vk_\gamma) \cdot e(-(vk_x + \pi_A + \pi_C), vk^2_{\beta\gamma}) e(-(vk^1_{\beta\gamma}), \pi_B) = 1$.

5. $e(vk_x + \pi_A, \pi_B) \cdot e(\pi_H, -vk_Z) \cdot e(\pi_C, -g_2) = 1$

Now pick $r_1, \ldots, r_5$ from a subset $S \subset \mathbb{F}$ of size s uniformly. We will check instead that a combination of the above factors with random powers is not 1; "shoving in" the exponents into the G_1 element of the pairing, we get the check

$$e(r_1 \cdot \pi_A, vk_A) \cdot e(r_1\pi'_A, -g_2) \cdot e(r_2 vk_B, \pi_B) \cdot e(r_2\pi'_B, -g_2) \cdot e(r_3\pi_C, vk_C) \cdot e(r_3\pi'_C, -g_2)$$

$$\cdot e(r_4\pi_K, vk_\gamma) \cdot e(-r_4(vk_x + \pi_A + \pi_C), vk^2_{\beta\gamma}) e(-r_4(vk^1_{\beta\gamma}), \pi_B) \cdot e(r_5(vk_x + \pi_A), \pi_B) \cdot e(r_5\pi_H, -vk_Z) \cdot e(r_5\pi_C, -g_2) = 1$$

Now, we merge together factors that have the same G_2 part, using the rule $e(a, c) \cdot e(b, c) = e(a + b, c)$. We get

$$e(r_1 \cdot \pi_A, vk_A) \cdot e(r_1\pi'_A + r_2\pi'_B + r_3\pi'_C + r_5\pi_C, -g_2) \cdot e(r_3\pi_C, vk_C) \cdot e(r_4\pi_K, vk_\gamma) \cdot e(-r_4(vk_x + \pi_A + \pi_C), vk^2_{\beta\gamma}) \cdot e(r_5\pi_H, -vk_Z$$

$$\cdot e(r_2 vk_B - r_4 vk^1_{\beta\gamma} + r_5(vk_x + \pi_A), \pi_B) = 1$$

2 Batch verification of proofs

Note that in 4 out of the 5 factors above, the G_2 argument depended only on the verification key. Thus, we can batch these factors from different proofs using accumulators.

1. a_1-accumulates the sum of $r_1\pi_A$
2. a_2-accumulates the sum of $r_1\pi'_A + r_2\pi'_B + r_3\pi'_C + r_5\pi_C$
3. a_3-accumulates the sum of $r_3\pi_C$
4. a_4-accumulates the sum of $r_4\pi_K$
5. a_5-accumulates the sum of $-r_4(vk_x + \pi_A + \pi_C)$
6. a_6-accumulates the sum of $r_5\pi_H$
7. a_7-accumulates the product of $ML(r_2 vk_B - r_4 vk^1_{\beta\gamma} + r_5(vk_x + \pi_A), \pi_B)$.

It is important to choose different $r_1, \ldots, r_5$ for each proof!

When the verifier is done accumulating proofs, and wants to check, probabilistically, if they are all valid. He computes

$$FE(ML(a_1, vk_A) \cdot ML(a_2, -g_2) \cdot ML(a_3, vk_C) \cdot ML(a_4, vk_\gamma) \cdot ML(a_5, vk^2_{\beta\gamma}) \cdot ML(a_6, -vk_Z) \cdot a_7) = 1.$$

In fact, one can save some time, by using a 6-fold Miller-Loop, to compute the product of the first 6 factors in the equation above.

One can show that a set of valid proofs will always be accepted, and a set of proof of which at least one is non-valid, will be accepted with probability at most $1/s$.