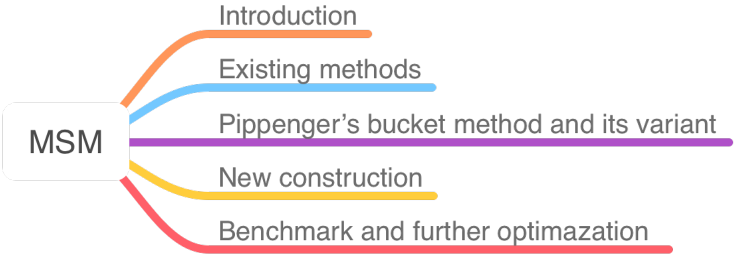
Speeding Up Multi-Scalar Multiplication Towards Efficient zkSNARKs

Guang Gong

Revised on June 9, 2022

UNIVERSITY OF WATERLOO

Outline


MSM
Introduction
Existing methods
Pippenger’s bucket method and its variant
New construction
Benchmark and further optimazation

Multi-scalar Multiplication

- Multi-scalar Multiplication (MSM) over fixed points:

$$S_{n,r} = a_1 P_1 + a_2 P_2 + ... + a_n P_n,\ 0 \leq a_i < r, P_i \in E. \quad (1)$$

Multi-scalar Multiplication

- Multi-scalar Multiplication (MSM) over fixed points:

$$S_{n,r} = a_1 P_1 + a_2 P_2 + \ldots + a_n P_n, \; 0 \le a_i < r, P_i \in E. \quad (1)$$

- MSM dominates the time consumption in the pairing-based trusted setup zkSNARKs.

Figure: credit: `https://en.wikipedia.org/wiki/Zcash`

Multi-scalar Multiplication

- Multi-scalar Multiplication (MSM) over fixed points:

$$S_{n,r} = a_1 P_1 + a_2 P_2 + \ldots + a_n P_n,\ 0 \leq a_i < r, P_i \in E. \quad (1)$$

- Example: Groth16 [Gro16],

  we may delete the groth16 formulas, they're long and complicated.

  proof computation: 3 MSM's,

$$A = G^{\alpha + \sum_{i=0}^{m} a_i u_i(x) + r\delta} \qquad B = H^{\beta + \sum_{i=0}^{m} a_i v_i(x) + s\delta}$$

$$C = G^{\frac{\sum_{i=\ell+1}^{m} a_i(\beta u_i(x) + \alpha v_i(x) + w_i(x)) + h(x)t(x)}{\delta} + s\left(\alpha + \sum_{i=0}^{m} a_i u_i(x)\right) + r\left(\beta + \sum_{i=0}^{m} a_i v_i(x)\right) + rs\delta}$$

  proof verification: 3 pairings and 1 MSM.

$$e(A, B) = e(G^{\alpha}, H^{\beta}) e(G^{\frac{\sum_{i=0}^{\ell} a_i\left(\beta u_i(x) + \alpha v_i(x) + w_i(x)\right)}{\gamma}}, H^{\gamma}) e(C, H^{\delta})$$

Multi-scalar Multiplication

- Multi-scalar Multiplication (MSM) over fixed points:

$$S_{n,r} = a_1P_1 + a_2P_2 + ... + a_nP_n,\ 0 \le a_i < r, P_i \in E. \quad (1)$$

- Example: Groth16 [Gro16],

  we may delete the groth16 formulas, they're long and complicated.

  proof computation: 3 MSM's,

$$A = G^{\alpha+\sum_{i=0}^{m} a_i u_i(x)+r\delta} \qquad B = H^{\beta+\sum_{i=0}^{m} a_i v_i(x)+s\delta}$$

$$C = G^{\frac{\sum_{i=\ell+1}^{m} a_i(\beta u_i(x)+\alpha v_i(x)+w_i(x))+h(x)t(x)}{\delta}+s\left(\alpha+\sum_{i=0}^{m} a_i u_i(x)\right)+r\left(\beta+\sum_{i=0}^{m} a_i v_i(x)\right)+rs\delta}$$

  proof verification: 3 pairings and 1 MSM.

$$e(A,B) = e(G^{\alpha}, H^{\beta})e(G^{\frac{\sum_{i=0}^{\ell} a_i(\beta u_i(x)+\alpha v_i(x)+w_i(x))}{\gamma}}, H^{\gamma})e(C, H^{\delta})$$

- Circuit size in Zcash: for hash, tens of thousands; for nested hash, several millions.

Existing methods

$S_{n,r}$ with small n.

- Binary method:
  doubling-and-addition,
  Knuth's 5 window algorithm [Knu97, BC89].

Existing methods

$S_{n,r}$ with small n.

- Binary method:
  doubling-and-addition,
  Knuth's 5 window algorithm [Knu97, BC89].
- Construction of number systems:
  basic digit sets [Mat82, BGMW92],
  multi-base number systems [DKS09, SIM12, YWLT13].

Existing methods

$S_{n,r}$ with small n.

- Binary method:
  doubling-and-addition,
  Knuth's 5 window algorithm [Knu97, BC89].
- Construction of number systems:
  basic digit sets [Mat82, BGMW92],
  multi-base number systems [DKS09, SIM12, YWLT13].
- Addition chains:
  PRAC chains [Mon92],
  DJB chains [Ber06],
  other multi-dimensional differential addition chains [Bro15, Rao15].

Existing methods

$S_{n,r}$ with small n.

- Binary method:
  doubling-and-addition,
  Knuth's 5 window algorithm [Knu97, BC89].
- Construction of number systems:
  basic digit sets [Mat82, BGMW92],
  multi-base number systems [DKS09, SIM12, YWLT13].
- Addition chains:
  PRAC chains [Mon92],
  DJB chains [Ber06],
  other multi-dimensional differential addition chains [Bro15, Rao15].
- Pippenger's bucket method and its variants.

Existing methods

When n is big ($2^{17} \leq n \leq 2^{23}$, or $10^5 \leq n \leq 10^7$).

- SOTA: Pippenger's bucket method and its variants.
- zkSNARK-oriented implementations, Zcash, TurboPLONK, Bellman, gnark, choose Pippenger's bucket method.

Pippenger's bucket method

- Example:

$$S_{13,8} = 2P_1 + 3P_2 + 7P_3 + 6P_4 + 5P_5 + 1P_6 + 3P_7 \\ +6P_8 + 2P_9 + 7P_{10} + 1P_{11} + 4P_{12} + 5P_{13}.$$

Pippenger's bucket method

- Example:

$$S_{13,8} = 2P_1 + 3P_2 + 7P_3 + 6P_4 + 5P_5 + 1P_6 + 3P_7 \\ +6P_8 + 2P_9 + 7P_{10} + 1P_{11} + 4P_{12} + 5P_{13}.$$

- All points are sorted into 7 buckets according to the scalars,

$$\begin{aligned} S_{13,8} =& 1 \cdot (P_6 + P_{11}) + 2 \cdot (P_1 + P_9) + 3 \cdot (P_2 + P_7) + 4 \cdot (P_{12}) \\ &+ 5 \cdot (P_5 + P_{13}) + 6 \cdot (P_4 + P_8) + 7 \cdot (P_3 + P_{10}) \\ :=& 1S_1 + 2S_2 + \ldots + 7S_7. \end{aligned}$$

Pippenger's bucket method

- Example:

$$S_{13,8} = 2P_1 + 3P_2 + 7P_3 + 6P_4 + 5P_5 + 1P_6 + 3P_7 \\ +6P_8 + 2P_9 + 7P_{10} + 1P_{11} + 4P_{12} + 5P_{13}.$$

- All points are sorted into 7 buckets according to the scalars,

$$\begin{aligned} S_{13,8} =& 1 \cdot (P_6 + P_{11}) + 2 \cdot (P_1 + P_9) + 3 \cdot (P_2 + P_7) + 4 \cdot (P_{12}) \\ & + 5 \cdot (P_5 + P_{13}) + 6 \cdot (P_4 + P_8) + 7 \cdot (P_3 + P_{10}) \\ :=& 1S_1 + 2S_2 + \ldots + 7S_7. \end{aligned}$$

The accumulated sum $\sum_{i=1}^{7} iS_i$ can be computed via

$$\begin{aligned} & S_7 \\ & + (S_7 + S_6) \\ & + (S_7 + S_6 + S_5) \\ & \ldots \\ & + (S_7 + S_6 + S_5 + \cdots + S_1). \end{aligned}$$

UNIVERSITY OF WATERLOO

Pippenger's bucket method

- Example:

$$\begin{aligned} S_{13,8} = {} & 2P_1 + 3P_2 + 7P_3 + 6P_4 + 5P_5 + 1P_6 + 3P_7 \\ & +6P_8 + 2P_9 + 7P_{10} + 1P_{11} + 4P_{12} + 5P_{13}. \end{aligned}$$

- All points are sorted into 7 buckets according to the scalars,

$$\begin{aligned} S_{13,8} = {} & 1 \cdot (P_6 + P_{11}) + 2 \cdot (P_1 + P_9) + 3 \cdot (P_2 + P_7) + 4 \cdot (P_{12}) \\ & + 5 \cdot (P_5 + P_{13}) + 6 \cdot (P_4 + P_8) + 7 \cdot (P_3 + P_{10}) \\ := {} & 1S_1 + 2S_2 + \ldots + 7S_7. \end{aligned}$$

- S_i's: $13 - 7 = 6$ additions,
  $\sum_{i=1}^{7} iS_i$: $2 \times 6 = 12$ additions.
  In total, 18 additions.

Pippenger's bucket method

- If r is small enough:

$$S_{n,r} = a_1P_1 + a_2P_2 + \cdots + a_nP_n.$$

- All points are sorted into $r-1$ buckets according to the scalars,

$$\begin{aligned} S_{n,r} =& 1S_1 + 2S_2 + \ldots + (r-1)S_{r-1} \\ =& S_{r-1} + (S_{r-1} + S_{r-2}) + \cdots + (S_{r-1} + S_{r-2} + \cdots S_1). \end{aligned}$$

- S_i's: $n-r+1$ additions,
  $\sum_{i=1}^{r-1} iS_i$: $2 \times (r-2)$ additions.
  In total, $n+r-3$ additions.

Pippenger's bucket method variant 1

- If r is big (over BLS12-381 curve, $r \approx 2^{256}$), every scalar is decomposed into q-ary form,

$$\begin{aligned} a_i &= a_{i0} + a_{i1}q + \cdots + a_{i,h-1}q^{h-1}. \\ S_{n,r} &= a_1P_1 + a_2P_2 + \cdots + a_nP_n \\ &= \sum_{i=1}^{n}\sum_{j=0}^{h-1} a_{ij} \cdot (q^jP_i), 0 \le a_{ij} < q, \\ &:= S_{nh,q}. \end{aligned}$$

Pippenger's bucket method variant 1

- If r is big (over BLS12-381 curve, $r \approx 2^{256}$), every scalar is decomposed into q-ary form,

$$\begin{aligned} a_i &= a_{i0} + a_{i1}q + \cdots + a_{i,h-1}q^{h-1}. \\ S_{n,r} &= a_1P_1 + a_2P_2 + \cdots + a_nP_n \\ &= \sum_{i=1}^{n}\sum_{j=0}^{h-1} a_{ij}\cdot(q^jP_i), 0 \le a_{ij} < q, \\ &:= S_{nh,q}. \end{aligned}$$

- Precomputation (nh Points):

$$\{q^jP_i \mid i = 1, 2, ..., n,\ j = 0, 1, 2, ..., h-1\},$$

- Using aforementioned method, all points are sorted into $q-1$ buckets, in total, $nh + q - 3$ additions [BGMW95].

Pippenger's bucket method variant 2

- Let us summary the framework of computing MSM,

$$S_{n,r} = S_{hn,q} = \sum_{i=1}^{n}\sum_{j=0}^{h-1} a_{ij} q^j P_i, 0 \leq a_i \leq q$$

- If $a_{ij} = m_{ij} b_{ij}$, $m_{ij} \in M$ (multiplier), $b_{ij} \in B$ (bucket),

$$S_{hn,q} = \sum_{i=1}^{n}\sum_{j=0}^{h-1} b_{ij} \cdot (m_{ij} q^j P_i).$$

Pippenger's bucket method variant 2

- Let us summary the framework of computing MSM,

$$S_{n,r} = S_{hn,q} = \sum_{i=1}^{n} \sum_{j=0}^{h-1} a_{ij} q^j P_i, 0 \leq a_i \leq q$$

- If $a_{ij} = m_{ij} b_{ij}$, $m_{ij} \in M$ (multiplier), $b_{ij} \in B$ (bucket),

$$S_{hn,q} = \sum_{i=1}^{n} \sum_{j=0}^{h-1} b_{ij} \cdot (m_{ij} q^j P_i).$$

- Precompute ($nh|M|$ points)
$\{mq^j P_i \mid 1 \leq i \leq n, 0 \leq j \leq h-1, m \in M\}$,
then it takes $\approx nh + |B|$ additions to compute $S_{n,r}$.

Pippenger's bucket method variant 2

- Let us summary the framework of computing MSM,

$$S_{n,r} = S_{hn,q} = \sum_{i=1}^{n} \sum_{j=0}^{h-1} a_{ij} q^j P_i, 0 \leq a_i \leq q$$

- If $a_{ij} = m_{ij} b_{ij}$, $m_{ij} \in M$ (multiplier), $b_{ij} \in B$ (bucket),

$$S_{hn,q} = \sum_{i=1}^{n} \sum_{j=0}^{h-1} b_{ij} \cdot (m_{ij} q^j P_i).$$

- Precompute ($nh|M|$ points)
  $\{mq^j P_i \mid 1 \leq i \leq n, 0 \leq j \leq h-1, m \in M\}$,
  then it takes $\approx nh + |B|$ additions to compute $S_{n,r}$.
- Pippenger's bucket method variant 1,
  $M = \{1\},\ B = \{0, 1, 2, ..., q-1\}$, it takes $\approx nh + q$ additions.

Pippenger's bucket method variant 2

- Let us summary the framework of computing MSM,

$$S_{n,r} = S_{hn,q} = \sum_{i=1}^{n} \sum_{j=0}^{h-1} a_{ij} q^j P_i, 0 \leq a_i \leq q$$

- If $a_{ij} = m_{ij} b_{ij}$, $m_{ij} \in M$ (multiplier), $b_{ij} \in B$ (bucket),

$$S_{hn,q} = \sum_{i=1}^{n} \sum_{j=0}^{h-1} b_{ij} \cdot (m_{ij} q^j P_i).$$

- Precompute ($nh|M|$ points)
  $\{mq^j P_i \mid 1 \leq i \leq n, 0 \leq j \leq h-1, m \in M\}$,
  then it takes $\approx nh + |B|$ additions to compute $S_{n,r}$.
- Pippenger's bucket method variant 2 (notice that $-P$ can be easily computed given P),
  $M = \{1, -1\}$, $B = \{0, 1, 2, ..., \lceil q/2 \rceil\}$, it takes $\approx nh + q/2$ additions.

UNIVERSITY OF WATERLOO

Question: Construct B, *s.t.* $|B| \approx q/\ell$?
then $S_{n,r}$ takes $\approx hn + q/\ell$ additions.

New construction

- (G.W. Luo, G. Gong, C.K. Weng) Let q be a prime *s.t.*
  2 is a primitive element in $\mathbb{F}_q$,
  ℓ and h be small positive integers *s.t.*
  $2^{\ell-1} < q$ and $q^{h-1} < r \leq 2^{\ell-1}q^{h-1}$.
- The multiplier set is

$$M = \{2^i \mid 0 \leq i \leq \ell - 1\} \cup \{-1\},$$

UNIVERSITY OF WATERLOO

New construction

- (G.W. Luo, G. Gong, C.K. Weng) Let q be a prime *s.t.* 2 is a primitive element in $\mathbb{F}_q$, ℓ and h be small positive integers *s.t.* $2^{\ell-1} < q$ and $q^{h-1} < r \leq 2^{\ell-1}q^{h-1}$.
- The multiplier set is

$$M = \{2^i \mid 0 \leq i \leq \ell - 1\} \cup \{-1\},$$

- The corresponding bucket set ($|B| \approx q/\ell$) is

$$B = \{i \mid 0 \leq i \leq 2^{\ell-1}\} \cup \{2^{i\cdot\ell} \bmod q \mid 0 \leq i \leq \lfloor q/\ell \rfloor\}.$$

The idea behind the construction is that

$$\{i \mid 1 \leq i < q\} = \{2^i \bmod q \mid 0 \leq i \leq q-2\}.$$

New construction

- (G.W. Luo, G. Gong, C.K. Weng) Let q be a prime *s.t.* 2 is a primitive element in $\mathbb{F}_q$, ℓ and h be small positive integers *s.t.* $2^{\ell-1} < q$ and $q^{h-1} < r \leq 2^{\ell-1}q^{h-1}$.
- The multiplier set is

$$M = \{2^i \mid 0 \leq i \leq \ell - 1\} \cup \{-1\},$$

- The corresponding bucket set ($|B| \approx q/\ell$) is

$$B = \{i \mid 0 \leq i \leq 2^{\ell-1}\} \cup \{2^{i\cdot\ell} \bmod q \mid 0 \leq i \leq \lfloor q/\ell \rfloor\}.$$

  The idea behind the construction is that

$$\{i \mid 1 \leq i < q\} = \{2^i \bmod q \mid 0 \leq i \leq q-2\}.$$

- Note: Elements in the bucket set is no longer consecutive, a new accumulation algorithm is needed.

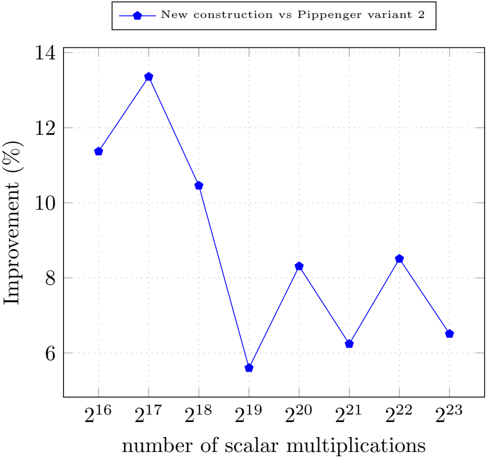
New construction

- Result: $S_{n,r}$ over fixed points can be computed using at most
$$\approx nh + q/\ell$$
additions, with the help of ℓnh precomputed points
$$\left\{ mq^j P_i \mid 1 \le i \le n, 0 \le j \le h-1, m \in \{1, 2, ..., 2^{\ell-1}\} \right\},$$
where $h = \lceil \log_q r \rceil$.
- For $\ell = 12$, the new construction saves 5% to 13% (depends on n) computational cost comparing to the most popular variant of Pippenger's bucket method.

Benchmark over BLS12-381 curve

n	ℓ	q	h	$\|B\|$	d	Number of additions
2^{16}	12	443413	14	40715	136	9.584×10^5
2^{17}	12	1310269	13	112939	125	1.817×10^6
2^{18}	12	1310269	13	112939	125	3.521×10^6
2^{19}	12	1310269	13	112939	125	6.929×10^6
2^{20}	12	4715027	12	396669	158	1.298×10^7
2^{21}	12	4715027	12	396669	158	2.556×10^7
2^{22}	12	21919501	11	1830350	173	4.797×10^7
2^{23}	12	21919501	11	1830350	173	9.411×10^7

Benchmark over BLS12-381 curve



UNIVERSITY OF WATERLOO

Further optimization

- Observation:

$$\{i \mid 1 \le i < q\} = \{2^i \bmod q \mid 0 \le i \le q-2\}$$
$$= \{2^i \bmod q \mid -(q-3)/2 \le i \le (q-1)/2\}.$$

Bucket set can be further reduced to ($|B| \approx q/(2\ell)$)

$$B = \left\{i \mid 0 \le i \le 2^\ell\right\} \cup \left\{2^{i\cdot\ell} \bmod q \mid 0 \le i \le \lfloor (q-1)/2\ell \rfloor\right\}$$

Further optimization

- Observation:

$$\{i \mid 1 \leq i < q\} = \{2^i \bmod q \mid 0 \leq i \leq q-2\}$$
$$= \{2^i \bmod q \mid -(q-3)/2 \leq i \leq (q-1)/2\}.$$

  Bucket set can be further reduced to ($|B| \approx q/(2\ell)$)

$$B = \left\{i \mid 0 \leq i \leq 2^\ell\right\} \cup \left\{2^{i\cdot\ell} \bmod q \mid 0 \leq i \leq \lfloor (q-1)/2\ell \rfloor\right\}$$

- GLV endomorphism

$$\lambda P = \lambda \cdot (x, y) = (\xi x, y),\ \lambda^3 = 1 \in \mathbb{F}_r, \xi^3 = 1 \in \mathbb{F}_p,$$

  $\lambda \approx \sqrt{r}$, every scalar $a = a_0 + a_1\lambda$, so

$$S_{n,r} = S_{2n,\lambda}.$$

  It further reduce the precomputation by a factor of 2.

Conclusion: $S_{n,r}$ over fixed points can be computed using at most

$$\approx 2n\lceil h/2 \rceil + q/(2\ell)$$

additions, with the help of $\approx \ell n \lceil h/2 \rceil$ precomputed points

$$\left\{ mq^j P_i \mid 1 \leq i \leq n, 0 \leq j \leq \lceil h/2 \rceil - 1, m \in \{1, 2, ..., 2^{\ell-1}\} \right\},$$

where $h = \lceil \log_q r \rceil$, and q is a prime selected to minimize the complexity.

UNIVERSITY OF WATERLOO

Happy Birthday to Prof. Doug Stinson!

Thanks