Title of the Presentation

Guiwen Luo

Department of Electrical and Computer Engineering
University of Waterloo
CANADA
guiwen.luo@uwaterloo.ca

insert Date

Joint work with xxx

Problem:

Multi-scalar Multiplication (MSM) over fixed points:

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

How can we compute it efficiently for large $n : n \geq 2^{10}$?

Outline

- Introduction
- Existing methods
- Pippenger's bucket method and its variant
- Our new construction
- Instantiation and experiment over BLS12-381 curve

Motivation

- MSM over fixed points dominates the time consumption in zero-knowledge succinct non-interactive argument of knowledge (zkSNARK) schemes with pairing-based trusted setup.
- Circuit size in Zcash: for single hash, SHA-256, the number of multiplication gates is about 23 thousands; for nested hash, several millions.

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

Existing methods

Computing $S_{n,r}$:

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

Existing methods

Computing $S_{n,r}$:

- **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].

UNIVERSITY OF WATERLOO

Existing methods

Computing $S_{n,r}$:

- **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

Computing $S_{n,r}$:

- **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 (cont.)

For large n ($n \geq 2^{10}$)

- 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:

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

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 $\{1, \cdots, 7\}$:

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

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 $\{1, \cdots, 7\}$:

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

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 $\{1, \cdots, 7\}$:

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

- $\{S_i\}$: $13 - 7 = 6$ 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:

$$S_{n,r} = a_1 P_1 + a_2 P_2 + \cdots + a_n P_n.$$

- 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

- If r is big (over BLS12-381 curve, $r \approx 2^{255}$), every scalar is decomposed into q-ary form,

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

Pippenger's bucket method variant

- If r is big (over BLS12-381 curve, $r \approx 2^{255}$), 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].

UNIVERSITY OF WATERLOO

Generalized framework

- 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),

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

Generalized framework

- 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 $S_{n,r}$.

UNIVERSITY OF WATERLOO

Generalized framework

- 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,
  $M = \{1\},\ B = \{0, 1, 2, ..., q-1\}$, it takes $\approx nh + q$ additions.

Generalized framework

- 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

Goal:

Construct B, *s.t.* $|B| \approx 0.21q$. Thus $S_{n,r}$ takes $\approx nh + 0.21q$ additions.

New construction

- (G.W. Luo, S.H. Fu, G. Gong) Let $q = 2^c$ be the radix. The multiplier set is $M = \{-3, -2, -1, 1, 2, 3\}$.

New construction

- (G.W. Luo, S.H. Fu, G. Gong) Let $q = 2^c$ be the radix. The multiplier set is $M = \{-3, -2, -1, 1, 2, 3\}$.
- Three auxiliary sets,

$$\begin{aligned} B_0 =& \{0\} \cup \{b \mid 1 \le b \le q/2, s.t.\ \omega_2(b) + \omega_3(b) \equiv 0 \bmod 2\}, \\ B_2 =& \{0\} \cup \{b \mid 1 \le b \le r_{h-1} + 1, s.t.\ \omega_2(b) + \omega_3(b) \equiv 0 \bmod 2\}, \end{aligned}$$

where $r_{h-1} = \lfloor r/q^{h-1} \rfloor$. B_1 is defined by the following algorithm

Input: B_0, q.
Output: B_1.

$\mathtt{B_1} = \mathtt{B_0}$
for $\mathtt{i} = \mathtt{q/4}$ to $\mathtt{q/2} - 1$ by 1 **do**
  if $\mathtt{i}$ is in $\mathtt{B_0}$ and $\mathtt{q} - 2 \cdot \mathtt{i}$ is in $\mathtt{B_0}$ **then**
    $\mathtt{B_1.remove}(\mathtt{q} - 2 \cdot \mathtt{i})$
for $\mathtt{i} = \lfloor \mathtt{q/6} \rfloor$ to $\mathtt{q/4} - 1$ by 1 **do**
  if $\mathtt{i}$ is in $\mathtt{B_0}$ and $\mathtt{q} - 3 \cdot \mathtt{i}$ is in $\mathtt{B_0}$ **then**
    $\mathtt{B_1.remove}(\mathtt{q} - 3 \cdot \mathtt{i})$
return $\mathtt{B_1}$

- The bucket set is constructed as $B = B_1 \cup B_2$.

UNIVERSITY OF WATERLOO

Comparison

$q = 2^c$ is the radix used to decompose the scalars, $h = \lceil \log_q r \rceil$. The time complexity of Pippenger's bucket set and Pippenger's variant hold if $r \leq q/2 \cdot q^{h-1}$. The time complexity of our construction holds when r/q^h is small.

Table: Comparison of different methods that computes $S_{n,r}$

Method	Storage	Complexity
Trivial method	$n \cdot P$	$3/2 \cdot (n \log_2 r) \cdot A$
Straus method [Str64]	$n2^c \cdot P$	$h(n + c) \cdot A$
Pippenger [Pip76, BDLO12]	$n \cdot P$	$h(n + q/2) \cdot A$
Pippenger variant [BGMW95]	$nh \cdot P$	$(nh + q/2) \cdot A$
Our construction [this work]	$3nh \cdot P$	$(nh + 0.21q) \cdot A$

UNIVERSITY OF WATERLOO

Instantiation and experiment over BLS12-381 curve

n	Pippenger	Pippenger variant	Our construction	Improv1	Improv2
2^{10}	3.69×10^4	2.46×10^4	2.22×10^4	39.8%	9.6%
2^{11}	6.66×10^4	4.51×10^4	4.23×10^4	36.4%	6.1%
2^{12}	1.20×10^5	8.60×10^4	8.12×10^4	32.2%	5.6%
2^{13}	2.21×10^5	1.64×10^5	1.49×10^5	32.4%	8.8%
2^{14}	4.06×10^5	2.95×10^5	2.80×10^5	30.8%	4.9%
2^{15}	7.37×10^5	5.57×10^5	5.43×10^5	26.4%	2.6%
2^{16}	1.39×10^6	1.08×10^6	1.03×10^6	26.3%	5.0%
2^{17}	2.62×10^6	2.10×10^6	1.92×10^6	26.6%	8.2%
2^{18}	4.72×10^6	3.93×10^6	3.63×10^6	23.1%	7.7%
2^{19}	8.91×10^6	7.34×10^6	7.04×10^6	21.1%	4.1%
2^{20}	1.73×10^7	1.42×10^7	1.35×10^7	22.2%	4.9%
2^{21}	3.30×10^7	2.73×10^7	2.60×10^7	21.2%	4.5%

UNIVERSITY OF WATERLOO

Instantiation and experiment over BLS12-381 curve

	$\mathbb{G}_1$		$\mathbb{G}_2$	
n	Improv1	Improv2	Improv1	Improv2
2^{10}	40.6%	8.86%	40.6%	9.26%
2^{11}	36.8%	6.54%	37.0%	6.78%
2^{12}	33.7%	5.78%	34.2%	5.74%
2^{13}	30.7%	4.40%	29.7%	3.13%
2^{14}	31.2%	6.54%	31.0%	7.29%
2^{15}	28.4%	3.19%	29.0%	3.61%
2^{16}	21.9%	-1.88%	21.8%	-2.05%
$*2^{16}$	24.6%	1.48%	24.9%	2.10%
2^{17}	22.1%	4.91%	21.6%	3.08%
2^{18}	22.8%	6.75%	23.0%	7.03%
2^{19}	20.3%	5.12%	20.8%	6.06%
2^{20}	17.7%	-0.13%	18.8%	1.45%
$*2^{20}$	19.4%	1.69%	17.9%	2.66%
2^{21}	19.0%	4.31%	–	–

UNIVERSITY OF WATERLOO

Conclusions:

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

$$\approx nh + 0.21q$$

additions, with the help of $3nh$ precomputed points

$$\left\{mq^j P_i \mid 1 \le i \le n, 0 \le j \le h-1, m \in \{1,2,3\}\right\},$$

where $q = 2^c$ is selected to minimize the complexity, $h = \lceil \log_q r \rceil$, r/q^h is small.

- Over BLS12-381 curve, when computing n-scalar multiplications for $n = 2^e$ $(10 \le e \le 21)$.
  - $21\%+$ improvement against Pippenger's bucket method.
  - 2.6% to 9.6% improvement against the variant [BGMW95].

UNIVERSITY OF WATERLOO