Reducing Complexity of Pairing Comparisons using Polynomial Evaluation

Adam Bobowski and Marcin Słowik

Department of Computer Science
at Faculty of Fundamental Problems of Technology
Wrocław University of Science and Technology
{first-name}.{last-name}@pwr.edu.pl

May 2018

Abstract

We propose a new method for reducing complexity of the pairing comparisons based on polynomials. Thought the construction introduces uncertainty into (usually deterministic) checks, it is easily quantifiable and in most cases extremely small. The application to CL-LRSW signature verification under n messages and group order q allows to reduce the number of computed pairings from $4n$ down to just 4, while the introduced uncertainty is just $\frac{2n-1}{q}$.

1 Introduction

Bilinear pairings is an interesting topic in cryptographic research. There is a great number of cryptographic protocols that use bilinear pairing comparisons as their core components and their versatility is without the question. The main issue, however, is their relative computational cost, which is rather high, compared to standard operations on elliptic curves. Due to their inefficiency they are not so widely adopted in the real world.

In this paper we propose a new and efficient way to significantly reduce the number and complexity of pairing comparisons, based on polynomial evaluation. The idea is to use sides of comparisons as polynomial coefficients, and compare the result of evaluation in randomly chosen point. Though the construction introduces some uncertainty into, usually deterministic, checks, the probability of failure is easily quantifiable. For practical applications, where the number of performed checks is negligible in comparison to the group order, our construction achieves a negligible false positive rate.

We later show an application of the method to the CL-LRSW signature scheme and results for a proof-of-concept implementation, in order to highlight the computational cost reduction.

1.1 Preliminaries

Definition (Real/False Positive/Negative) *In this paper, we introduce randomness and uncertainty to, usually deterministic, checks. The uncertainty is quantified by probability of certain events. However, given that our guarantees are dependent on the* values to be checked *to be correct or incorrect, we cannot simply declare one, universal value.*

To quantify the quality of our proposal, we measure the chance of a false-positive *and* false-negative *event. There are two additional, complimentary events, that can be defined using the aforementioned values:* real-positive *and* real-negative*.*

In cryptography, false-positive *is usually associated with* soundness *property (what is the chance that someone without proper input can simulate the protocol), whilst* real-positive *is usually associated with* completeness *property (what is the chance that given proper inputs, the protocol succeeds).*

Let $\mathcal{A}$ denote the output of our algorithm, and let $\mathcal{O}$ denote an output of an "ideal" algorithm. In cases presented in this paper, the "ideal" algorithm corresponds to a routine comparison on value-by-value basis.

Output 1 means "verification passed" and 0 means "verification failed".

$$\begin{aligned}
\Pr[\textsf{false-positive}] &= \Pr[1 \leftarrow \mathcal{A} | 0 \leftarrow \mathcal{O}] \\
\Pr[\textsf{false-negative}] &= \Pr[0 \leftarrow \mathcal{A} | 1 \leftarrow \mathcal{O}] \\
\Pr[\textsf{real-positive}] &= \Pr[1 \leftarrow \mathcal{A} | 1 \leftarrow \mathcal{O}] \\
\Pr[\textsf{real-negative}] &= \Pr[0 \leftarrow \mathcal{A} | 0 \leftarrow \mathcal{O}]
\end{aligned}$$

Note that

$$\begin{aligned}
\Pr[\textsf{real-positive}] &= 1 - \Pr[\textsf{false-negative}] \\
\Pr[\textsf{real-negative}] &= 1 - \Pr[\textsf{false-positive}].
\end{aligned}$$

Definition (Bilinear Maps (Pairings)) *Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two (additive in notation) groups, containing subgroups of prime order q. We say that e is a bilinear map, if there exist another (multiplicative in notation) cyclic group $\mathbb{G}_T$ of order q, and a pairing function $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ such that:*

1. *(Bilinear) For all quadruples of elements $(A, B, C, D) \in \mathbb{G}_1^2 \times \mathbb{G}_2^2$, the function is bilinear, that is:*

$$e(A + B, C + D) = e(A, C) \cdot e(A, D) \cdot e(B, C) \cdot e(B, D).$$

2. *(Non-degenerate) There exist a pair of inputs $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$, that creates a non-trivial solution, that is $e(P, Q) \neq 1$, where 1 is the identity of $\mathbb{G}_T$.*

3. *(Efficient) There exists an efficient algorithm for computing e.*

2 Efficient comparison via polynomial evaluation

Let us first consider a generic case of comparing pairwise multiple pairs of elements of cyclic groups. Assume that, for some reason, the comparison itself is expensive, e.g. elements are located at remote hosts, or are coset representatives and have to be normalized.

Theorem 1 *Let $\mathbb{G}$ be a cyclic group of order q and let g be its generator. Given a series of n pairs $(L_i, R_i) \in \mathbb{G}^2$, a sequential comparison*

$$L_i \overset{?}{=} R_i \quad \forall i \in \{0, ...n-1\}$$

is roughly equivalent to

$$\prod_{i=0}^{n-1} L_i^{x^i} \overset{?}{=} \prod_{i=0}^{n-1} R_i^{x^i}$$

for $x \in \mathbb{Z}_q$ chosen uniformly at random, up to a probability of false positive not greater than $\frac{n-1}{q}$.

Proof *Since $\mathbb{G}$ is cyclic, each element L_i and R_i can be represented as a power of the generator g, that is:*

$$L_i = g^{\mu_i} \qquad\qquad R_i = g^{\nu_i}.$$

Therefore, for each $i \in \{0, \ldots n-1\}$:

$$L_i = R_i \iff \mu_i \equiv \nu_i \mod q.$$

Consider the following polynomials, $\mu(X)$ and $\nu(X)$:

$$\mu(X) = \sum_{i=0}^{n-1} \mu_i X^i \qquad\qquad \nu(X) = \sum_{i=0}^{n-1} \nu_i X^i$$

We know that if for some x

$$\begin{aligned} \mu(x) &= \nu(x) \\ \mu(x) - \nu(x) &= 0 \\ (\mu - \nu)(x) &= 0, \end{aligned}$$

then x is either a solution to a non-trivial polynomial $(\mu - \nu)(X) = \sum_{i=0}^{n-1}(\mu_i - \nu_i)(X)$, or all coefficients $\mu_i = \nu_i$. We know that a polynomial of degree $n-1$ has at most $n-1$ solutions. Thus, for x chosen uniformly at random, the probability of (blindly) finding a solution of the polynomial is bounded by:

$$Pr\big[(\alpha - \beta)(x) = 0 \wedge \alpha(X) \neq \beta(X)\big] \leq \frac{n-1}{q}.$$

We can write:

$$\begin{aligned} g^{\mu(x)} &= g^{\sum_{i=0}^{n-1} \mu_i x^i} & g^{\nu(x)} &= g^{\sum_{i=0}^{n-1} \nu_i x^i} \\ &= \prod_{i=0}^{n-1} L_i^{x^i} & &= \prod_{i=0}^{n-1} R_i^{x^i} \end{aligned}$$

Therefore, instead of comparing the equalities for a sequence of value pairs, we can compare the products

$$\prod_{i=0}^{n-1} L_i^{x^i} \overset{?}{=} \prod_{i=0}^{n-1} R_i^{x^i}$$

for a uniformly chosen value x. If the sequences are identical, the result is correct, thus $\Pr[\textit{false-negative}] = 0$. In case of inequalities, we obtain the incorrect result with probability $\Pr[\textit{false-positive}] \leq \frac{n-1}{q}$. $\square$

In this paper, we restrict the case to cyclic groups of prime order q, because of the natural isomorphism between them and $\mathbb{F}_q$. The construction works in many other cases, as long as a similar isomorphism (not necessarily efficiently computable) to a ring can be found.

Note that this construction works for all cases of cyclic groups, regardless of any hardness assumptions. The false positive rate is purely statistical and depends on uniform selection of the challenge x. There are no external parties, no adversaries, etc.

3 Application to Bilinear Maps

The same principle can be applied to bilinear maps, with additional advantages.

Theorem 2 *Assume $\mathbb{G}_1$ and $\mathbb{G}_2$ are pairing preimage groups, and $\mathbb{G}_T$ is a pairing target group with order q. Given a series of n quadruples*

$$(A_i, B_i, C_i, D_i) \in \mathbb{G}_1^2 \times \mathbb{G}_2^2,$$

a sequential comparisons of bilinear maps

$$e(A_i, C_i) \stackrel{?}{=} e(B_i, D_i) \quad \forall i \in \{0, ...n-1\}$$

where each $A_i, C_i \in \mathbb{G}_1$ and $B_i, D_i \in \mathbb{G}_2$, for $x \in \mathbb{Z}_q$ chosen uniformly at random, yields the same output as

$$\prod_{i=0}^{n-1} e(A_i, C_i)^{x^i} \stackrel{?}{=} \prod_{i=0}^{n-1} e(B_i, D_i)^{x^i}$$

with $\Pr[\textit{false-negative}] = 0$ and $\Pr[\textit{false-positive}] \leq \frac{n-1}{q}$.

Proof *Let us denote $L_i = e(A_i, C_i)$ and $R_i = e(B_i, D_i)$. Given that both L_i and R_i are elements of a cyclic group $\mathbb{G}_T$, Theorem 1 holds.* □

Corollary 1 *Using bilinear properties, Theorem 2 can be used to reduce the number of pairing function computations, if there are common elements between the input quadruples $(A_i, B_i, C_i, D_i) \in \mathbb{G}_1^2 \times \mathbb{G}_2^2$.*

Example *Consider a series of n checks of a form $e(G, A_i) \stackrel{?}{=} e(H, B_i)$ for $i \in \{0, ...n-1\}$. Using Theorem 2, the series can be replaced with a single check:*

$$\prod_{i=0}^{n-1} e(G, A_i)^{x^i} \stackrel{?}{=} \prod_{i=0}^{n-1} e(H, B_i)^{x^i}$$

However, using bilinear property of e, this can be rewritten, without loss of generality, as:

$$\prod_{i=0}^{n-1} e(G, [x^i]A_i) \stackrel{?}{=} \prod_{i=0}^{n-1} e(H, [x^i]B_i)$$

$$e\left(G, \sum_{i=0}^{n-1} [x^i]A_i\right) \stackrel{?}{=} e\left(H, \sum_{i=0}^{n-1} [x^i]B_i\right)$$

This procedure reduced the total number of e calls from $2n$ to 2.

Note that there are no assumptions on $\mathbb{G}_1$ or $\mathbb{G}_2$ being cyclic, or having any other properties, apart from being preimage groups for a non-degenerate bilinear pairing.

4 Application to CL-LRSW Signatures

While the technique has numerous applications, we only present a few of them. Notably, the reduction can be used to simplify verification of the CL-LRSW [CL04] signatures by a significant factor. Using the technique, we reduced number of expensive pairing operations from $4l+4$ down to 4, on the cost of $4l+2$ (much cheaper) $\mathbb{G}_1/\mathbb{G}_2$ scalar multiplications and extremely small probability of false-positive outcome.

Corollary 2 (CL-LRSW Reduction) *Let $\sigma = (A_0, \ldots A_l, B_0, \ldots B_l, C)$ be a CL-LRSW signature (Scheme C) under $l+1$ messages m_i and let $\mathsf{pk} = (X^{(2)}, Y^{(2)}, Z_1^{(2)}, \ldots Z_l^{(2)})$ be a valid CL-LRSW public key (where the index ${}^{(2)}$ indicates the element belongs to $\mathbb{G}_2$). Standard sequential verification can be reduced to just 4 pairing operations, on the cost of $4l+2$ $\mathbb{G}_1/\mathbb{G}_2$ scalar multiplications and $\Pr[\textsf{false-positive}] = \frac{2l+1}{q}$.*

```
StdVerify:
1 :  $e(C, G_2) \overset{?}{=} e(A_0, X^{(2)}) \cdot \prod_{i=0}^{l} e([m_i]B_i, X^{(2)})$
2 :  $e(A_0, Z_i^{(2)}) \overset{?}{=} e(A_i, G_2) \quad \forall i \in \{1, \ldots l\}$
3 :  $e(A_i, Y^{(2)}) \overset{?}{=} e(B_i, G_2) \quad \forall i \in \{0, \ldots l\}$
```

```
PolyVerify:
1 :  $x \leftarrow_\$ \mathbb{Z}_q$
2 :  $L = e(C + \sum_{i=1}^{l}[x^i]A_i + \sum_{i=0}^{l}[x^{l+i+1}]B_i, G_2)$
3 :  $R = e(A_0 + \sum_{i=0}^{l}[m_i]B_i, X^{(2)}) \cdot e(A_0, \sum_{i=1}^{l}[x^i]Z_i^{(2)}) \cdot e(\sum_{i=0}^{l}[x^{l+i+1}]A_i, Y^{(2)})$
4 :  $L \overset{?}{=} R$
```

Proof *Let us rearrange and label the equations as follows:*

$$\begin{aligned} L_0 &= e(C, G_2) & R_0 &= e(A_0 + \textstyle\sum_{i=0}^{l}[m_i]B_i, X^{(2)}) \\ L_i &= e(A_i, G_2) & R_i &= e(A_0, Z_i^{(2)}) \quad \forall i \in \{1, \ldots l\} \\ L_{i+l+1} &= e(B_i, G_2) & R_{i+l+1} &= e(A_i, Y^{(2)}) \quad \forall i \in \{0, \ldots l\}. \end{aligned}$$

From Theorem 2 with $n = 2l+2$, it follows that for a standard polynomial verification, the false positive rate is bounded by $\Pr\big[\textsf{false-positive}\big] \le \frac{2l+1}{q}$. In a similar fashion to Corollary 1, the pairing functions can be folded to obtain more compact form:

$$\begin{cases} L = e(C, G_2)^{x^0} \cdot \prod_{i=1}^{l} e(A_i, G_2)^{x^i} \cdot \prod_{i=0}^{l} e(B_i, G_2)^{x^{l+i+1}} \\ R = e(A_0 + \sum_{i=0}^{l} [m_i] B_i, X^{(2)})^{x^0} \cdot \prod_{i=1}^{l} e(A_0, Z_i^{(2)})^{x^i} \cdot \prod_{i=0}^{l} e(A_i, Y^{(2)})^{x^{l+i+1}} \end{cases}$$

$$\begin{cases} L = e(C, G_2) \cdot e(\sum_{i=1}^{l} [x^i] A_i, G_2) \cdot e(\sum_{i=0}^{l} [x^{l+i+1}] B_i, G_2) \\ R = e(A_0 + \sum_{i=0}^{l} [m_i] B_i, X^{(2)}) \cdot e(A_0, \sum_{i=1}^{l} [x^i] Z_i^{(2)}) \cdot e(\sum_{i=0}^{l} [x^{l+i+1}] A_i, Y^{(2)}) \end{cases}$$

$$\begin{cases} L = e(C + \sum_{i=1}^{l} [x^i] A_i + \sum_{i=0}^{l} [x^{l+i+1}] B_i, G_2) \\ R = e(A_0 + \sum_{i=0}^{l} [m_i] B_i, X^{(2)}) \cdot e(A_0, \sum_{i=1}^{l} [x^i] Z_i^{(2)}) \cdot e(\sum_{i=0}^{l} [x^{l+i+1}] A_i, Y^{(2)}) \end{cases}$$

Given that all cases are equivalent, PolyVerify *also satisfies Theorem 2 with* $n = 2l + 2$*, thus for a simplified polynomial verification, the false positive rate is still bounded by* $\Pr[\textsf{false-positive}] \leq \frac{2l+1}{q}$. □

The same technique works with interactive protocols proving possession of an LRSW signature and its message. Given the compact formulae from [SW17], the checks can be translated nearly 1-to-1.

Corollary 3 (CL-LRSW ZKP Reduction) *Let* $\sigma' = (\widetilde{A}_0, \dots \widetilde{A}_l, \widetilde{B}_0, \dots \widetilde{B}_l, \widetilde{C})$, $T, c, s_r, s_0, \dots s_l$ *be the inputs to the verification step of the* Improved Verification Protocol *from [SW17], and* $\mathsf{pk} = (X^{(2)}, Y^{(2)}, Z_1^{(2)}, \dots Z_l^{(2)})$ *be a valid CL-LRSW public key.*

Standard sequential verification can be reduced to just 4 *pairing operations. On the cost of* $4l + 2$ $\mathbb{G}_1/\mathbb{G}_2$ *scalar multiplications and* $\Pr[\textsf{false-positive}] = \frac{2l+1}{q}$.

```
StdVerify:
1 :  $e(\widetilde{A}_0, Z_i^{(2)}) \overset{?}{=} e(\widetilde{A}_i, G_2) \quad \forall i \in \{1, \dots l\}$
2 :  $e(\widetilde{A}_i, Y^{(2)}) \overset{?}{=} e(\widetilde{B}_i, G_2) \quad \forall i \in \{0, \dots l\}$
3 :  $e([c]\widetilde{C}, G_2) \overset{?}{=} e(T - [s_r]\widetilde{A}_0 - \sum_{i=0}^{l} [s_i]\widetilde{B}_i, X^{(2)})$
```

```
PolyVerify:
1 :  $x \leftarrow_{\$} \mathbb{Z}_q$
2 :  $L = e([c]\widetilde{C} + \sum_{i=1}^{l} [x^i]\widetilde{A}_i + \sum_{i=0}^{l} [x^{l+i+1}]\widetilde{B}_i, G_2)$
3 :  $R = e(T - [s_r]\widetilde{A}_0 - \sum_{i=0}^{l} [s_i]\widetilde{B}_i, X^{(2)}) \cdot e(\widetilde{A}_0, \sum_{i=1}^{l} [x^i] Z_i^{(2)}) \cdot e(\sum_{i=0}^{l} [x^{l+i+1}]\widetilde{A}_i, Y^{(2)})$
4 :  $L \overset{?}{=} R$
```

Proof *The corollary is trivial and can be proven identically as Corollary 2.* □

5 Practical Results

To verify if the results of proposed modification are in any case practical, we have created a proof-of-concept implementation of CL-LRSW verification using Python library Charm-Crypto. Tests were run on MacBook 12 (2016). We considered multiple scenarios where number of signed messages $n = l - 1$ differs. Average time of verification is presented in the Table 1, and the visualization of the difference on Figure 1.

No. of messages	3	10	20	30	40	50
Standard	0.5715	1.6732	3.3550	5.0331	6.6795	8.3960
Polynomial	0.1905	0.2578	0.3505	0.4474	0.5411	0.6316

Table 1: Verification time (in seconds) of CL-LRSW on BN254 Elliptic Curve

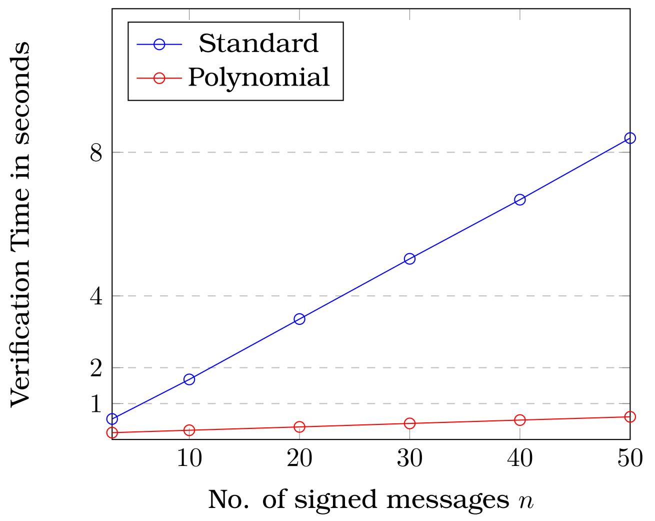

Figure 1: Verification Time of CL-LRSW on BN254 Elliptic Curve

The results meet the expectations, the verification with polynomials hugely improves the execution time.

6 Acknowledgement

This research was partially supported by the Polish National Science Centre (NCN) under grant OPUS no 2014/15/B/ST6/02837.

References

[CL04] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Annual International Cryptology Conference*, pages 56–72. Springer, 2004.

[SW17] Marcin Slowik and Marta Wszola. An efficient verification of CL-LRSW signatures and a pseudonym certificate system. In *Proceedings of the 4th ACM International Workshop on ASIA Public-Key Cryptography*, APKC '17, pages 13–23, New York, NY, USA, 2017. ACM, ACM.