

Cryptanalysis of the generalised Legendre pseudorandom function

Novak Kaluderovic, Thorsten Kleinjung, Dusan Kostic

July 2, 2020

EPFL

Background

Pseudorandom functions

Legendre pseudorandom functions

Pseudorandom functions

Set of functions $\{\mathcal{O}_k\}_k$ with the same domain and codomain such that a function chosen randomly over the set of keys k cannot be distinguished from a random function.

Legendre pseudorandom functions

Pseudorandom functions

Set of functions $\{\mathcal{O}_k\}_k$ with the same domain and codomain such that a function chosen randomly over the set of keys k cannot be distinguished from a random function.

Legendre symbol

Legendre pseudorandom functions

Pseudorandom functions

Set of functions $\{\mathcal{O}_k\}_k$ with the same domain and codomain such that a function chosen randomly over the set of keys k cannot be distinguished from a random function.

Legendre symbol

Let p be an odd prime.

$$\left(\frac{x}{p}\right) = \begin{cases} 1 & \text{if } x \in \mathbb{F}_p \text{ is a square} \\ -1 & \text{if } x \in \mathbb{F}_p \text{ is not a square.} \end{cases}$$

Legendre pseudorandom functions

Pseudorandom functions

Set of functions $\{\mathcal{O}_k\}_k$ with the same domain and codomain such that a function chosen randomly over the set of keys k cannot be distinguished from a random function.

Legendre symbol

Let p be an odd prime.

$$\left(\frac{x}{p}\right) = \begin{cases} 1 & \text{if } x \in \mathbb{F}_p \text{ is a square} \\ -1 & \text{if } x \in \mathbb{F}_p \text{ is not a square.} \end{cases}$$

$$\left(\frac{a}{p}\right) \left(\frac{b}{p}\right) = \left(\frac{ab}{p}\right).$$

Damgård, 1988 [Dam90]: The Legendre PRF

Damgård, 1988 [Dam90]: The Legendre PRF

$$\mathcal{O}_k(x) = \left(\frac{x+k}{p} \right), \quad k \in \mathbb{F}_p$$

Damgård, 1988 [Dam90]: The Legendre PRF

$$\mathcal{O}_k(x) = \left(\frac{x+k}{p} \right), \quad k \in \mathbb{F}_p$$

Russell, Shparlinski, 2004 [RS04]: The Generalised Legendre PRF

Damgård, 1988 [Dam90]: The Legendre PRF

$$\mathcal{O}_k(x) = \left(\frac{x+k}{p} \right), \quad k \in \mathbb{F}_p$$

Russell, Shparlinski, 2004 [RS04]: The Generalised Legendre PRF

$$\mathcal{O}_f(x) = \left(\frac{f(x)}{p} \right), \quad f \in \mathbb{F}_p[x]_r$$

Orders of magnitude slower than cryptographic PRFs.

Orders of magnitude slower than cryptographic PRFs.

Grassi et al., 2016 [GRR⁺16]: Suitable for multiparty computation.

Orders of magnitude slower than cryptographic PRFs.

Grassi et al., 2016 [GRR⁺16]: Suitable for multiparty computation.

Ethereum, 2019 [Fei19]: Plans to incorporate it in the blockchain.

Orders of magnitude slower than cryptographic PRFs.

Grassi et al., 2016 [GRR⁺16]: Suitable for multiparty computation.

Ethereum, 2019 [Fei19]: Plans to incorporate it in the blockchain.

Ethereum, 2019 [Fei19]: Online challenges to break the function.

Legendre Sequences

Legendre Sequences

Legendre sequence

Let $a \in \mathbb{F}_p$ and $L \in \mathbb{N}$,

$$\{a\}_L := \left(\frac{a}{p}\right), \left(\frac{a+1}{p}\right), \left(\frac{a+2}{p}\right), \dots, \left(\frac{a+L-1}{p}\right).$$

Legendre Sequences

Legendre sequence

Let $a \in \mathbb{F}_p$ and $L \in \mathbb{N}$,

$$\{a\}_L := \left(\frac{a}{p}\right), \left(\frac{a+1}{p}\right), \left(\frac{a+2}{p}\right), \dots, \left(\frac{a+L-1}{p}\right).$$

Assumption

For $L = 2\lfloor \log p \rfloor$ we have

$$\{a\}_L = \{b\}_L \text{ if and only if } a = b.$$

Legendre Sequences

Generalised Legendre sequence

Let $f \in \mathbb{F}_p[x]_r$ and $L \in \mathbb{N}$,

$$\{f\}_L := \left(\frac{f(0)}{p}\right), \left(\frac{f(1)}{p}\right), \left(\frac{f(2)}{p}\right), \dots, \left(\frac{f(L-1)}{p}\right).$$

Generalised assumption:

For $L = 2\lfloor r \log p \rfloor$ we have

$$\{f\}_L = \{g\}_L \text{ if and only if } f = g.$$

Algorithm

Problem: Given access to $\mathcal{O}_f(x) = (\frac{f(x)}{p})$, find f .

Algorithm

Problem: Given access to $\mathcal{O}_f(x) = (\frac{f(x)}{p})$, find f .

Solution: Table-based collision search

Algorithm

Problem: Given access to $\mathcal{O}_f(x) = \left(\frac{f(x)}{p}\right)$, find f .

Solution: Table-based collision search

Table:

Make a table with many Legendre sequences $\{f_m\}_L$ such that

Problem: Given access to $\mathcal{O}_f(x) = \left(\frac{f(x)}{p}\right)$, find f .

Solution: Table-based collision search

Table:

Make a table with many Legendre sequences $\{f_m\}_L$ such that

- The sequence $\{f_m\}_L$ can be computed from \mathcal{O}_f .

Problem: Given access to $\mathcal{O}_f(x) = \left(\frac{f(x)}{p}\right)$, find f .

Solution: Table-based collision search

Table:

Make a table with many Legendre sequences $\{f_m\}_L$ such that

- The sequence $\{f_m\}_L$ can be computed from \mathcal{O}_f .
- From f_m we can obtain f .

Problem: Given access to $\mathcal{O}_f(x) = \left(\frac{f(x)}{p}\right)$, find f .

Solution: Table-based collision search

Table:

Make a table with many Legendre sequences $\{f_m\}_L$ such that

- The sequence $\{f_m\}_L$ can be computed from \mathcal{O}_f .
- From f_m we can obtain f .

Search:

Generate random $g(x)$ and look for $\{g\}_L$ in the table.

Algorithm

Problem: Given access to $\mathcal{O}_f(x) = \left(\frac{f(x)}{p}\right)$, find f .

Solution: Table-based collision search

Table:

Make a table with many Legendre sequences $\{f_m\}_L$ such that

- The sequence $\{f_m\}_L$ can be computed from \mathcal{O}_f .
- From f_m we can obtain f .

Search:

Generate random $g(x)$ and look for $\{g\}_L$ in the table.

If $\{g\}_L = \{f_m\}_L$ then $g = f_m$, and we can obtain f .

Möbius transformations

Rational transformations of \mathbb{P}^1 :

$$\begin{aligned}\varphi_m : \mathbb{P}^1 &\longrightarrow \mathbb{P}^1 \\ [x : y] &\longmapsto [ax + by : cx + dy],\end{aligned}$$

Möbius transformations

Rational transformations of \mathbb{P}^1 :

$$\begin{aligned}\varphi_m : \mathbb{P}^1 &\longrightarrow \mathbb{P}^1 \\ [x : y] &\longmapsto [ax + by : cx + dy],\end{aligned}$$

Isomorphic to $PGL_2(\mathbb{F}_p)$ given by $\varphi_m \leftrightarrow m = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

Möbius transformations

Rational transformations of \mathbb{P}^1 :

$$\begin{aligned}\varphi_m : \mathbb{P}^1 &\longrightarrow \mathbb{P}^1 \\ [x : y] &\longmapsto [ax + by : cx + dy],\end{aligned}$$

Isomorphic to $PGL_2(\mathbb{F}_p)$ given by $\varphi_m \leftrightarrow m = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

Action on monic polynomials:

Möbius transformations

Rational transformations of \mathbb{P}^1 :

$$\begin{aligned}\varphi_m : \mathbb{P}^1 &\longrightarrow \mathbb{P}^1 \\ [x : y] &\longmapsto [ax + by : cx + dy],\end{aligned}$$

Isomorphic to $PGL_2(\mathbb{F}_p)$ given by $\varphi_m \leftrightarrow m = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

Action on monic polynomials:

$$m \cdot f(x) = f_m(x)$$

Möbius transformations

Rational transformations of \mathbb{P}^1 :

$$\begin{aligned}\varphi_m : \mathbb{P}^1 &\longrightarrow \mathbb{P}^1 \\ [x : y] &\longmapsto [ax + by : cx + dy],\end{aligned}$$

Isomorphic to $PGL_2(\mathbb{F}_p)$ given by $\varphi_m \leftrightarrow m = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

Action on monic polynomials:

$$m \cdot f(x) = f_m(x) := f\left(\frac{ax+b}{cx+d}\right)$$

Möbius transformations

Rational transformations of \mathbb{P}^1 :

$$\begin{aligned}\varphi_m : \mathbb{P}^1 &\longrightarrow \mathbb{P}^1 \\ [x : y] &\longmapsto [ax + by : cx + dy],\end{aligned}$$

Isomorphic to $PGL_2(\mathbb{F}_p)$ given by $\varphi_m \leftrightarrow m = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

Action on monic **polynomials**:

$$m \cdot f(x) = f_m(x) := f\left(\frac{ax+b}{cx+d}\right)$$

Möbius transformations

Rational transformations of \mathbb{P}^1 :

$$\begin{aligned}\varphi_m : \mathbb{P}^1 &\longrightarrow \mathbb{P}^1 \\ [x : y] &\longmapsto [ax + by : cx + dy],\end{aligned}$$

Isomorphic to $PGL_2(\mathbb{F}_p)$ given by $\varphi_m \leftrightarrow m = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

Action on monic polynomials:

$$m \cdot f(x) = f_m(x) := f\left(\frac{ax+b}{cx+d}\right)(cx+d)^r$$

Möbius transformations

Rational transformations of \mathbb{P}^1 :

$$\begin{aligned}\varphi_m : \mathbb{P}^1 &\longrightarrow \mathbb{P}^1 \\ [x : y] &\longmapsto [ax + by : cx + dy],\end{aligned}$$

Isomorphic to $PGL_2(\mathbb{F}_p)$ given by $\varphi_m \leftrightarrow m = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

Action on **monic** polynomials:

$$m \cdot f(x) = f_m(x) := f\left(\frac{ax+b}{cx+d}\right)(cx+d)^r$$

Möbius transformations

Rational transformations of \mathbb{P}^1 :

$$\begin{aligned}\varphi_m : \mathbb{P}^1 &\longrightarrow \mathbb{P}^1 \\ [x : y] &\longmapsto [ax + by : cx + dy],\end{aligned}$$

Isomorphic to $PGL_2(\mathbb{F}_p)$ given by $\varphi_m \leftrightarrow m = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

Action on monic polynomials:

$$m \cdot f(x) = f_m(x) := f\left(\frac{ax+b}{cx+d}\right)(cx+d)^r / \left(f\left(\frac{a}{c}\right)c^r\right)$$

$$f_m(x) = f\left(\frac{ax+b}{cx+d}\right)(cx+d)^r / \left(f\left(\frac{a}{c}\right)c^r\right).$$

Möbius transformations

$$f_m(x) = f\left(\frac{ax+b}{cx+d}\right)(cx+d)^r / \left(f\left(\frac{a}{c}\right)c^r\right).$$

Computing $\{f_m\}_L$ from \mathcal{O}_f :

$$\left(\frac{f_m(x)}{p}\right) = \mathcal{O}_f\left(\frac{ax+b}{cx+d}\right) \left(\frac{cx+d}{p}\right)^r \mathcal{O}_f\left(\frac{a}{c}\right) \left(\frac{c}{p}\right)^r.$$

Möbius transformations

$$f_m(x) = f\left(\frac{ax+b}{cx+d}\right)(cx+d)^r / \left(f\left(\frac{a}{c}\right)c^r\right).$$

Computing $\{f_m\}_L$ from \mathcal{O}_f :

$$\left(\frac{f_m(x)}{p}\right) = \mathcal{O}_f\left(\frac{ax+b}{cx+d}\right) \left(\frac{cx+d}{p}\right)^r \mathcal{O}_f\left(\frac{a}{c}\right) \left(\frac{c}{p}\right)^r.$$

Linear cost per sequence: $L + 1$ oracle queries and $L + 1$ Legendre symbol computations \rightarrow 1 Legendre sequence.

Möbius transformations

$$f_m(x) = f\left(\frac{ax+b}{cx+d}\right)(cx+d)^r / (f\left(\frac{a}{c}\right)c^r).$$

Computing $\{f_m\}_L$ from \mathcal{O}_f :

$$\left(\frac{f_m(x)}{p}\right) = \mathcal{O}_f\left(\frac{ax+b}{cx+d}\right) \left(\frac{cx+d}{p}\right)^r \mathcal{O}_f\left(\frac{a}{c}\right) \left(\frac{c}{p}\right)^r.$$

Linear cost per sequence: $L + 1$ oracle queries and $L + 1$ Legendre symbol computations \rightarrow 1 Legendre sequence.

Amortised for all $m \in PGL_2(\mathbb{F}_p)$: p oracle queries and p Legendre symbols $\rightarrow (p^3 - p)$ Legendre sequences.

Precomputation

Create a table T containing $\{f_m\}_L$ for all $m \in PGL_2(\mathbb{F}_p)$.

In total $p^3 - p$ sequences.

Precomputation

Create a table T containing $\{f_m\}_L$ for all $m \in PGL_2(\mathbb{F}_p)$.

In total $p^3 - p$ sequences.

Search

Sample random $g(x)$ of degree r and compute $\{g\}_L$ until a hit is found.

Precomputation

Create a table T containing $\{f_m\}_L$ for all $m \in PGL_2(\mathbb{F}_p)$.

In total $p^3 - p$ sequences.

Search

Sample random $g(x)$ of degree r and compute $\{g\}_L$ until a hit is found.

Expected run-time: $O(p^{r-3})$ trials.

Limited query and the linear prf

Run-time: $\tilde{O}(p^3 + p^{r-3})$

Limited query and the linear prf

Run-time: $\tilde{O}(p^3 + p^{r-3})$

For $r < 6$ can be lowered to $\tilde{O}(p^{r/2} + p^{r/2})$ by limiting the table.

Limited query and the linear prf

Run-time: $\tilde{O}(p^3 + p^{r-3})$

For $r < 6$ can be lowered to $\tilde{O}(p^{r/2} + p^{r/2})$ by limiting the table.

Oracle queries needed : p

Limited query and the linear prf

Run-time: $\tilde{O}(p^3 + p^{r-3})$

For $r < 6$ can be lowered to $\tilde{O}(p^{r/2} + p^{r/2})$ by limiting the table.

Oracle queries needed : $p - o(p/L)$.

Limited query and the linear prf

Run-time: $\tilde{O}(p^3 + p^{r-3})$

For $r < 6$ can be lowered to $\tilde{O}(p^{r/2} + p^{r/2})$ by limiting the table.

Oracle queries needed : $p - o(p/L)$.

What if oracle queries are limited?

Limited query and the linear prf

How many different group actions can we obtain from $M \ll p$ queries?

Limited query and the linear prf

How many different group actions can we obtain from $M \ll p$ queries?

Linear shifts subgroup

Limited query and the linear prf

How many different group actions can we obtain from $M \ll p$ queries?

Linear shifts subgroup

$$G = \left\{ \begin{pmatrix} d & i \\ 0 & 1 \end{pmatrix} \middle| d \in \mathbb{F}_p^*, i \in \mathbb{F}_p \right\} \leq \mathrm{PGL}_2(\mathbb{F}_p).$$

Limited query and the linear prf

How many different group actions can we obtain from $M \ll p$ queries?

Linear shifts subgroup

$$G = \left\{ \begin{pmatrix} d & i \\ 0 & 1 \end{pmatrix} \middle| d \in \mathbb{F}_p^*, i \in \mathbb{F}_p \right\} \leq \mathrm{PGL}_2(\mathbb{F}_p).$$

$$\begin{pmatrix} d & i \\ 0 & 1 \end{pmatrix} \cdot f = f_{i,d}(x) = f(dx + i)/d^r.$$

Limited query and the linear prf

How many different group actions can we obtain from $M \ll p$ queries?

Linear shifts subgroup

$$G = \left\{ \begin{pmatrix} d & i \\ 0 & 1 \end{pmatrix} \middle| d \in \mathbb{F}_p^*, i \in \mathbb{F}_p \right\} \leq \mathrm{PGL}_2(\mathbb{F}_p).$$

$$\begin{pmatrix} d & i \\ 0 & 1 \end{pmatrix} \cdot f = f_{i,d}(x) = f(dx + i)/d^r.$$

$$\left(\frac{f_{i,d}(x)}{p} \right) = \mathcal{O}_f(dx + i) \left(\frac{d}{p} \right)^r.$$

Example

Query \mathcal{O}_f at $[0, M)$.

Example

Query \mathcal{O}_f at $[0, M)$.



Example

Query \mathcal{O}_f at $[0, M)$.



Example

Query \mathcal{O}_f at $[0, M)$.



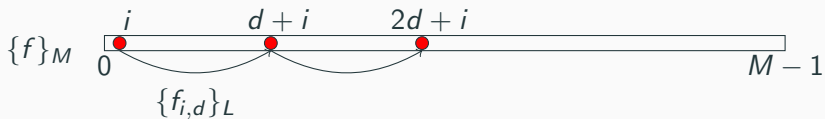
Example

Query \mathcal{O}_f at $[0, M)$.



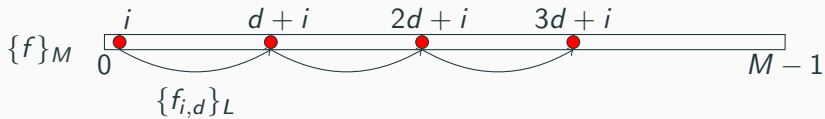
Example

Query \mathcal{O}_f at $[0, M)$.



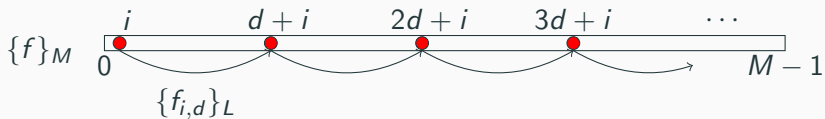
Example

Query \mathcal{O}_f at $[0, M)$.



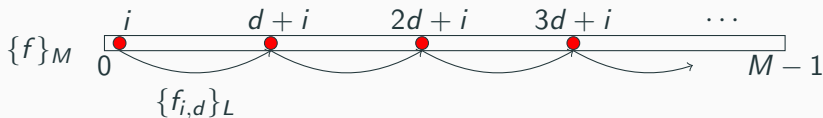
Example

Query \mathcal{O}_f at $[0, M)$.



Example

Query \mathcal{O}_f at $[0, M)$.



In total $\frac{M^2}{L}$ eligible (i, d) values.

Precomputation

Query \mathcal{O}_f at $[0, M)$. Make a table T with $O(\frac{M^2}{L})$ sequences.

Precomputation

Query \mathcal{O}_f at $[0, M)$. Make a table T with $O(\frac{M^2}{L})$ sequences.

Search

Try random polynomials until a hit is found in the table.

Precomputation

Query \mathcal{O}_f at $[0, M)$. Make a table T with $O(\frac{M^2}{L})$ sequences.

Search

Try random polynomials until a hit is found in the table.

Expected run-time: $O(\frac{p^r L}{M^2})$ trials.

Comparison

Khovratovich [Kho19]: Group G with $d = 1$. Table size: $O(1)$.

Khovratovich [Kho19]: Group G with $d = 1$. Table size: $O(1)$.

Beullens et al. [BBUV19]: Group G with $i < d$. Table size $\frac{M^2}{L^2}$.

Comparison

Khovratovich [Kho19]: Group G with $d = 1$. Table size: $O(1)$.

Beullens et al. [BBUV19]: Group G with $i < d$. Table size $\frac{M^2}{L^2}$.

Us: Full group G . Table size $\frac{M^2}{L}$.

Comparison

Khovratovich [Kho19]: Group G with $d = 1$. Table size: $O(1)$.

Beullens et al. [BBUV19]: Group G with $i < d$. Table size $\frac{M^2}{L^2}$.

Us: Full group G . Table size $\frac{M^2}{L}$.

Algorithm	expected # trials	precomputation	memory
Khovratovich	$\frac{p \log p}{M}$	M	$\log p$
Beullens et al.	$\frac{p \log^2 p}{M^2}$	M^2	$\frac{M^2}{\log p}$
Our algorithm	$\frac{p \log p}{M^2}$	$\frac{M^2}{\log p}$	M^2

Khovratovich [Kho19]: Group G with $d = 1$. Table size: $O(1)$.

Khovratovich [Kho19]: Group G with $d = 1$. Table size: $O(1)$.

Beullens et al. [BBUV19]: Group G with $i < d$. Table size $\frac{p^2}{L^2}$.

Khovratovich [Kho19]: Group G with $d = 1$. Table size: $O(1)$.

Beullens et al. [BBUV19]: Group G with $i < d$. Table size $\frac{p^2}{L^2}$.

Us: Full group $PGL_2(\mathbb{F}_p)$. Table size $p^3 - p$.

General case

Khovratovich [Kho19]: Group G with $d = 1$. Table size: $O(1)$.

Beullens et al. [BBUV19]: Group G with $i < d$. Table size $\frac{p^2}{L^2}$.

Us: Full group $PGL_2(\mathbb{F}_p)$. Table size $p^3 - p$.

Algorithm	search	precomputation	memory
Khovratovich	$p^{r-1} r \log p$	$r \log p$	$r \log p$
Beullens et al.	$p^{r-2} r^2 \log^2 p$	p^2	p^2
Our algorithm	$p^{r-3} r \log p$	$p^3 r \log p$	$p^3 r \log p$

Ethereum research challenges [Fei19]:

Ethereum research challenges [Fei19]:

- 5 Linear Legendre PRF challenges

Ethereum research challenges [Fei19]:

- 5 Linear Legendre PRF challenges
- Primes p of 64, 74, 84, 100 and 148 bits

Challenges

Ethereum research challenges [Fei19]:

- 5 Linear Legendre PRF challenges
- Primes p of 64, 74, 84, 100 and 148 bits
- Given $M = 2^{20}$ symbols of sequence $\{f\}_M$.

Ethereum research challenges [Fei19]:

- 5 Linear Legendre PRF challenges
- Primes p of 64, 74, 84, 100 and 148 bits
- Given $M = 2^{20}$ symbols of sequence $\{f\}_M$.
- Goal to find $f = x + k$.

Challenges

Ethereum research challenges [Fei19]:

- 5 Linear Legendre PRF challenges
- Primes p of 64, 74, 84, 100 and 148 bits
- Given $M = 2^{20}$ symbols of sequence $\{f\}_M$.
- Goal to find $f = x + k$.
- For each challenge we used $L = 64$.

Challenges

Ethereum research challenges [Fei19]:

- 5 Linear Legendre PRF challenges
- Primes p of 64, 74, 84, 100 and 148 bits
- Given $M = 2^{20}$ symbols of sequence $\{f\}_M$.
- Goal to find $f = x + k$.
- For each challenge we used $L = 64$.
- Tables contained 2^{34} sequences.

Challenges

Ethereum research challenges [Fei19]:

- 5 Linear Legendre PRF challenges
- Primes p of 64, 74, 84, 100 and 148 bits
- Given $M = 2^{20}$ symbols of sequence $\{f\}_M$.
- Goal to find $f = x + k$.
- For each challenge we used $L = 64$.
- Tables contained 2^{34} sequences.
- About 2.2e6 trials per core-second.

Results




Table 1: Results and estimates for solving the Legendre PRF challenges.
In all cases $M = 2^{20}$ consecutive queries are given.

Challenge	Prime bit size	Expected # trials	Observed # trials	Expected core-hours	Observed core-hours
0	64	2^{30}	$2^{30.78}$	290 sec	490 sec
1	74	2^{40}	$2^{39.53}$	82	59
2	84	2^{50}	$2^{46.97}$	1.4e5	1.72e4
3	100	2^{66}	-	9.1e9	-
4	148	2^{114}	-	2.5e24	-

The end

Thank you for Your attention!

-  Ward Beullens, Tim Beyne, Aleksei Udovenko, and Giuseppe Vitto, *Cryptanalysis of the Legendre PRF and generalizations*, Cryptology ePrint Archive, Report 2019/1357, 2019, <https://eprint.iacr.org/2019/1357>.
-  Ivan Damgård, *On the randomness of Legendre and Jacobi sequences*, Proceedings of the 8th Annual International Cryptology Conference on Advances in Cryptology (London, UK), CRYPTO '88, Springer-Verlag, 1990, pp. 163–172.
-  Dankard Feist, *Legendre pseudo-random function*, 2019, <https://legendreprf.org/bounties>.

-  Lorenzo Grassi, Christian Rechberger, Dragos Rotaru, Peter Scholl, and Nigel P. Smart, *MPC-friendly symmetric key primitives*, Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (New York, NY, USA), CCS '16, ACM, 2016, pp. 430–443.
-  Dmitry Khovratovich, *Key recovery attacks on the Legendre PRFs within the birthday bound*, Cryptology ePrint Archive, Report 2019/862, 2019, <https://eprint.iacr.org/2019/862>.
-  Alexander Russell and Igor E. Shparlinski, *Classical and quantum function reconstruction via character evaluation*, Journal of Complexity **20** (2004), no. 2-3, 404–422 (English).