Cryptanalysis of the Legendre PRF and Generalizations

W. Beullens[1] T. Beyne[1] A. Udovenko[2] G. Vitto[2]

[1]imec-COSIC, ESAT, KULeuven

[2]SnT, University of Luxembourg

November 13, 2020

Legendre symbol

- Legendre symbol of $a \in \mathbb{F}_p$ (prime $p > 2$):

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{if } a = b^2 \text{ for some } b \in \mathbb{F}_p^\times, \\ 0 & \text{if } a = 0, \\ -1 & \text{otherwise.} \end{cases}$$

Legendre symbol

- Legendre symbol of $a \in \mathbb{F}_p$ (prime $p > 2$):

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{if } a = b^2 \text{ for some } b \in \mathbb{F}_p^\times, \\ 0 & \text{if } a = 0, \\ -1 & \text{otherwise.} \end{cases}$$

- Early 1900s: equidistribution results
  Jacobsthal (1906) and Davenport (1931)

- Damgård (1990) conjectures pseudorandomness of

$$\left(\frac{k}{p}\right), \left(\frac{k+1}{p}\right), \ldots$$

Legendre PRF

- Pseudorandom function proposed by Grassi et al. (2016):

$$L_k(x) = \left(\frac{x+k}{p}\right) \in \{-1, 0, 1\}$$

- MPC-friendly
- Applications
  - Ethereum 2.0 proof-of-custody
  - LegRoast signatures Beullens et al. (2020)

Cryptanalysis of the Legendre PRF

Overview

- 24/07 — $\widetilde{\mathcal{O}}(p/M)$
  Khovratovich (2019)
- 20/08 — Bounties announced
- 09/10 — Challenges announced ($M = 2^{20}$)

(Time complexities for $M < \sqrt[4]{p}$.)

Cryptanalysis of the Legendre PRF

Overview

24/07 — $\widetilde{\mathcal{O}}(p/M)$
Khovratovich (2019)

20/08 — Bounties announced

09/10 — Challenges announced ($M = 2^{20}$)

https://legendreprf.org/

Concrete instances

At Devcon5, further bounties for concrete instances of the Legendre PRF were announced. For primes of size 64–148 (security levels 24–108[2]), the following bounties are now available for recovering a Legendre key:

Prime size	Security	Prize	
64 bits	24 bits	1 ETH	CLAIMED
74 bits	34 bits	2 ETH	CLAIMED
84 bits	44 bits	4 ETH	CLAIMED
100 bits	60 bits	8 ETH	
148 bits	108 bits	16 ETH	

For each of the challenges, 2^{20} bits of output from the Legendre PRF are available here. To claim one of these bounties, you must find the correct key that generates the outputs.

(Time complexities for $M < \sqrt[4]{p}$.)

Cryptanalysis of the Legendre PRF

Overview

- 24/07 — $\widetilde{\mathcal{O}}(p/M)$ Khovratovich (2019)
- 20/08 — Bounties announced
- 09/10 — Challenges announced ($M = 2^{20}$)
- 14/10 — 64-bit challenge solved
- 21/10 — 74-bit challenge solved

$\widetilde{\mathcal{O}}(p/M^2)$ this work

- 30/11 — 84-bit challenge solved
  $\log p$ improvement by Kaluđerović et al. (2020)

https://legendreprf.org/

Concrete instances

At Devcon5, further bounties for concrete instances of the Legendre PRF were announced. For primes of size 64–108 (security levels 24–108[2]), the following bounties are now available for recovering a Legendre key:

Prime size	Security	Prize	
64 bits	24 bits	1 ETH	CLAIMED
74 bits	34 bits	2 ETH	CLAIMED
84 bits	44 bits	4 ETH	CLAIMED
100 bits	60 bits	8 ETH	
148 bits	108 bits	16 ETH	

For each of the challenges, 2^{20} bits of output from the Legendre PRF are available here. To claim one of these bounties, you must find the correct key that generates the outputs.

(Time complexities for $M < \sqrt[4]{p}$.)

Cryptanalysis of the Legendre PRF
Khovratovich (2019)

- Notation: $L_k(x + [m]) = (L_k(x), L_k(x+1), \ldots, L_k(x+m-1))$

Cryptanalysis of the Legendre PRF

Khovratovich (2019)

- Notation: $L_k(x + [m]) = (L_k(x), L_k(x+1), \ldots, L_k(x+m-1))$

- Observation: $L_k(x + [m]) = L_0(k + x + [m])$

$\vdots$	$\vdots$
a	$L_k(a + [m])$
$\vdots$	$\vdots$

1. Query $L_k([M])$
2. Extract $M - m$ sequences of the form $L_k(a + [m])$

Sample $L_0(c + [m])$ until collision
if $m = \Omega(\log p)$ then
probably $c = k + a$

Cryptanalysis of the Legendre PRF

Khovratovich (2019)

- Notation: $L_k(x+[m]) = (L_k(x), L_k(x+1), \dots, L_k(x+m-1))$

- Observation: $L_k(x+[m]) = L_{0}(k+x+[m])$

$\vdots$	$\vdots$
a	$L_k(a+[m])$
$\vdots$	$\vdots$

1. Query $L_k([M])$
2. Extract $M-m$ sequences of the form $L_k(a+[m])$

Sample $L_0(c+[m])$ until collision
if $m = \Omega(\log p)$ then
probably $c = k + a$

Cost: $\widetilde{\mathcal{O}}(M + p/M)$ operations
$\widetilde{\mathcal{O}}(M)$ memory

Cryptanalysis of the Legendre PRF

Our attack: idea

- Multiplicativity of the Legendre symbol:

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right) \implies L_0(b)\, L_{k/b}(a/b+[m]) = L_k(a+b[m])$$

Cryptanalysis of the Legendre PRF

Our attack: idea

- Multiplicativity of the Legendre symbol:

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right) \implies L_0(b)\, L_{k/b}(a/b+[m]) = L_k(a+b[m])$$

$\vdots$	$\vdots$
a, b	$L_{k/b}(a/b+[m])$
$\vdots$	$\vdots$

1. Query $L_k([M])$
2. Extract $\sim M^2/m$ sequences of the form $L_{k/b}(a/b+[m])$

Sample $L_0(c+[m])$ until collision
if $m = \Omega(\log p)$ then
probably $c = (k+a)/b$

Cryptanalysis of the Legendre PRF

Our attack: idea

- Multiplicativity of the Legendre symbol:

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right) \implies L_0(b)\, L_{k/b}(a/b+[m]) = L_k(a+b[m])$$

$\vdots$	$\vdots$
a, b	$L_{k/b}(a/b+[m])$
$\vdots$	$\vdots$

1. Query $L_k([M])$
2. Extract $\sim M^2/m$ sequences of the form $L_{k/b}(a/b+[m])$

Sample $L_0(c+[m])$ until collision
if $m = \Omega(\log p)$ then
probably $c = (k+a)/b$

Cost: $\widetilde{\mathcal{O}}(M^2 + p/M^2)$ operations
$\mathcal{O}(M^2)$ memory

Cryptanalysis of the Legendre PRF

Our attack: optimizations

- Use consecutive samples in offline phase:

  1. Compute $L_0(c + [w])$ for some $w > m$

  2. Extract $\sim w^2/m$ sequences of the form $L_0(c/d + [m])$

Cryptanalysis of the Legendre PRF

Our attack: optimizations

- Use consecutive samples in offline phase:

  1. Compute $L_0(c + [w])$ for some $w > m$
  2. Extract $\sim w^2/m$ sequences of the form $L_0(c/d + [m])$

- *Caveat*: sequences in the table are not random
- Advantages:
  - Amortizes Legendre symbol computation
    $\rightarrow$ Cost dominated by sequence extraction and table lookups
  - Only store sequences with $|a| < |b|$
- Cost: $\mathcal{O}(M^2 + p \log^2 p / M^2)$ time
  $\mathcal{O}(M^2 / \log p)$ memory

Cryptanalysis of the Legendre PRF

Our attack: implementation results

- First $M = 2^{20}$ consecutive PRF outputs $L_k([M])$ were given
- Bottleneck: table lookups ($0.08 \mu s$)

p	Time (core-hours)	Memory / thread (GB)
$2^{40} - 87$	$<$ 0.001	$<$ 1
$2^{64} - 59$	1.5	3
$2^{74} - 35$	1500	3

- Dell C6420 server; two Intel Xeon Gold 6132 CPUs (2.6 GHz)
  128 GB of RAM

`https://github.com/cryptolu/LegendrePRF`

Generalizations of the Legendre PRF
Overview

- Higher-degree Legendre PRF
  First analysis by Khovratovich (2019)

- Jacobi symbols
- Power-residue symbols

} Damgård (1990)

Generalizations of the Legendre PRF

Higher-degree Legendre PRF

- Degree-1 Legendre PRF:

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

Generalizations of the Legendre PRF
Higher-degree Legendre PRF

- Degree-d Legendre PRF:

$$L_k(x) = \left(\frac{x^d + k_{d-1}x^{d-1} + \ldots + k_1 x + k_0}{p} \right), \quad k \in \mathbb{F}_p^d$$

Generalizations of the Legendre PRF
Higher-degree Legendre PRF

- Degree-d Legendre PRF:

$$L_k(x) = \left(\frac{x^d + k_{d-1}x^{d-1} + \ldots + k_1 x + k_0}{p}\right), \quad k \in \mathbb{F}_p^d$$

- Attacks ($d \geq 2$):
  - Khovratovich (2019): $\widetilde{\mathcal{O}}(p^{d-1})$ time
  - This work: $\widetilde{\mathcal{O}}(p^2 + p^{d-2})$ using sequence extraction
  - Kaluđerović et al. (2020): $\widetilde{\mathcal{O}}(p^3 + p^{d-3})$
  - Weak keys (next slides)

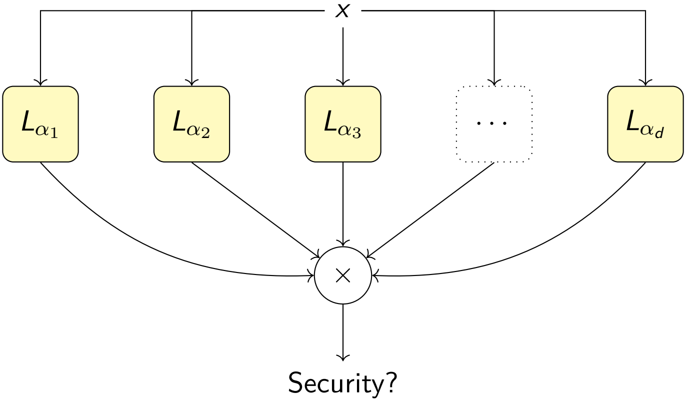
Generalizations of the Legendre PRF

Higher-degree Legendre PRF

- Example:

$$x^d + k_{d-1}x^{d-1} + \ldots + k_1 x + k_0 = \prod_{i=1}^{d}(x - \alpha_i)$$

with $\alpha_1, \ldots, \alpha_d \in \mathbb{F}_p$ distinct



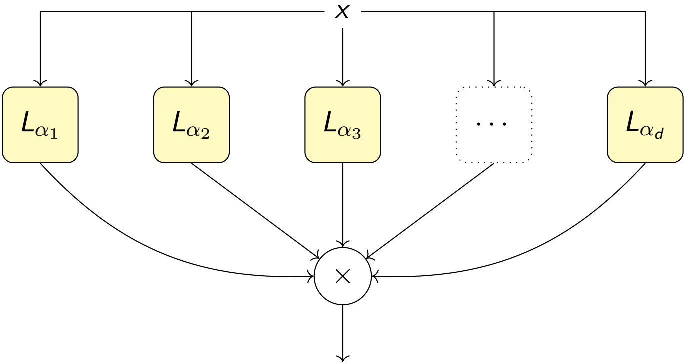
Generalizations of the Legendre PRF

Higher-degree Legendre PRF

- Example:

$$x^d + k_{d-1}x^{d-1} + \ldots + k_1 x + k_0 = \prod_{i=1}^{d}(x - \alpha_i)$$

with $\alpha_1, \ldots, \alpha_d \in \mathbb{F}_p$ distinct



Security?

✖ $\widetilde{\mathcal{O}}(p^{\lceil d/2 \rceil})$ attack

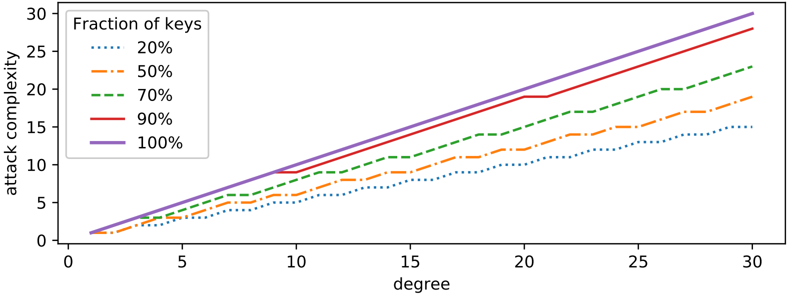
Generalizations of the Legendre PRF

Higher-degree Legendre PRF

- Weak key when $x^d + k_{d-1}x^{d-1} + \ldots + k_1 x + k_0$ is reducible
- Worst case: two factors of equal degree

$$L_k(x) = L_{k_1}(x) L_{k_2}(x) \text{ with } k_1, k_2 \in \mathbb{F}_p^{d/2}$$

- Attack: find collision between $L_k([m])L_{k_1}([m])$ and $L_{k_2}([m])$



Generalizations of the Legendre PRF

Jacobi PRF

- Let $p, q > 2$ be primes. Jacobi symbol of $a \in \mathbb{Z}/(pq)\mathbb{Z}$:

$$\left(\frac{a}{pq}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{q}\right)$$

Generalizations of the Legendre PRF
Jacobi PRF

- Let $p, q > 2$ be primes. Jacobi symbol of $a \in \mathbb{Z}/(pq)\mathbb{Z}$:

$$\left(\frac{a}{pq}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{q}\right)$$

- Observation

$$\left(\frac{k+px}{pq}\right) = \left(\frac{k}{p}\right)\left(\frac{k+px}{q}\right) = \left(\frac{k}{p}\right)\left(\frac{p}{q}\right)\left(\frac{k/p+x}{q}\right)$$

- Attack:

  1. Use attack on Legendre PRF to obtain $k \bmod q$
  2. Use attack on Legendre PRF to obtain $k \bmod p$
  3. Apply the Chinese Remainder Theorem

Generalizations of the Legendre PRF

Power-residue PRF

- Let p be a prime such that $r \mid (p-1)$

- The r-th power residue symbol of x is

$$\left(\frac{x}{p}\right)_r = x^{(p-1)/r}$$

- Applications
  - Extract more output-bits
  - PorcRoast signatures Beullens et al. (2020)

Generalizations of the Legendre PRF

Power-residue PRF

- Let p be a prime such that $r \mid (p-1)$

- The r-th power residue symbol of x is

$$\left(\frac{x}{p}\right)_r = x^{(p-1)/r}$$

- Applications
  - Extract more output-bits
  - PorcRoast signatures Beullens et al. (2020)

- Basic attack generalizes: $\widetilde{\mathcal{O}}(M^2 + p/M^2)$ time
  $\widetilde{\mathcal{O}}(M^2)$ memory

- For large r: $\widetilde{\mathcal{O}}(M + p/(Mr))$ time $\widetilde{\mathcal{O}}(M)$ memory (see paper)

Conclusions

- Improved attack on the Legendre PRF
  - Relevant in the low-data setting: $\widetilde{\mathcal{O}}(p/M^2)$ for $M < \sqrt[4]{p}$
  - Solution to concrete challenges (64 and 74 bit)
- Improved attacks on the higher-degree variant
- First evaluation of two other variants from Damgård (1990)
  - Jacobi symbols
  - Power-residue symbols

https://github.com/cryptolu/LegendrePRF

References I

- Jacobsthal, Ernst Erich (1906). “Anwendungen einer Formel aus der Theorie der quadratischen Reste”. PhD thesis. Friedrich-Wilhelms Universität zu Berlin. Chap. 3.
- Davenport, Harold (1931). “On the distribution of quadratic residues (mod p)”. In: *Journal of the London Mathematical Society* 1.1, pp. 49–54.
- Damgård, Ivan (Aug. 1990). “On the Randomness of Legendre and Jacobi Sequences”. In: *CRYPTO’88*. Ed. by Shafi Goldwasser. Vol. 403. LNCS. Springer, Heidelberg, pp. 163–172. DOI: 10.1007/0-387-34799-2_13.
- Grassi, Lorenzo et al. (Oct. 2016). “MPC-Friendly Symmetric Key Primitives”. In: *ACM CCS 2016*. Ed. by Edgar R. Weippl et al. ACM Press, pp. 430–443. DOI: 10.1145/2976749.2978332.
- Beullens, Ward et al. (2020). *LegRoast: Efficient post-quantum signatures from the Legendre PRF*. Cryptology ePrint Archive, Report 2020/128. https://eprint.iacr.org/2020/128.

References II

Khovratovich, Dmitry (2019). *Key recovery attacks on the Legendre PRFs within the birthday bound*. Cryptology ePrint Archive, Report 2019/862. https://eprint.iacr.org/2019/862.

Kaluđerović, Novak et al. (2020). *Improved key recovery on the Legendre PRF*. Cryptology ePrint Archive, Report 2020/098. https://eprint.iacr.org/2020/098.