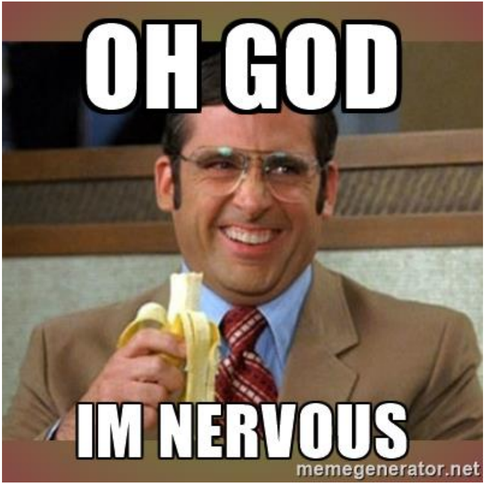
Efficient Implementation of Huff Curve

N. Gamze Orhon

Department of Computer Engineering
Yasar University

June 2017
Summerschool on Real-world Crypto and Privacy


OH GOD
IM NERVOUS
memegenerator.net

Who am I?

Bachelor

- Yasar University Software Engineering 2009-2014

MSc

- Yasar University Computer Engineering 2014-

PhD

- ??????????????????????????????????????

mailto:*gamze@ngorhon.com*

visit:*ngorhon.com*

MSc Thesis

Aim

- To improve the efficiency of Huff curve $y(1+ax^2) = cx(1+dy^2)$

Methods

- $\mathbb{P}^1 \times \mathbb{P}^1$ embedding
- 2-isogeny decomposition

Outcome

- Faster group operations on Huff form.

Extended Huff Curve

Curve model	h	**DBL**	**muADD**	**uADD**
Wu, Feng, $\mathbb{P}^2$, $X(aY^2 - Z^2) = Y(bX^2 - Z^2)$	4	6**M**+5**S**+1**D**	10**M**+1**D**	11**M**+1**D**
Joye, Tibouchi, Vergnaud, $\mathbb{P}^2$, $aX(Y^2 - Z^2) = bY(X^2 - Z^2)$	8	6**M**+5**S**	10**M**	11**M**
This work, $\mathbb{P}^1 \times \mathbb{P}^1$, $YT(Z^2 + 2X^2) = cXZ(T^2 + 2Y^2)$	4	8**M** 4×2**M**	8**M** 4×2**M**	10**M**

Embedding

Embed Huff curve in

$\mathbb{P}^2$

or

$\mathbb{P}^1 \times \mathbb{P}^1$?

Embedding

Addition formulas for $\mathbb{P}^2$:

$$\Big((X_1Z_2 + X_2Z_1)(Z_1Z_2 + aX_1X_2)(Z_1Z_2 - dY_1Y_2)^2 : \\ (Y_1Z_2 + Y_2Z_1)(Z_1Z_2 + dY_1Y_2)(Z_1Z_2 - aX_1X_2)^2 : \\ (Z_1^2Z_2^2 - a^2X_1^2X_2^2)(Z_1^2Z_2^2 - d^2Y_1^2Y_2^2) \Big)$$

Embedding

Addition formulas for $\mathbb{P}^1 \times \mathbb{P}^1$:

$$\Big(\big((X_1Z_2 + Z_1X_2)(T_1T_2 - dY_1Y_2) : (Z_1Z_2 - aX_1X_2)(T_1T_2 + dY_1Y_2)\big), \\ \big((Z_1Z_2 - aX_1X_2)(Y_1T_2 + T_1Y_2) : (Z_1Z_2 + aX_1X_2)(T_1T_2 - dY_1Y_2)\big) \Big)$$

Embedding

Each coordinate of the point addition formulas in $\mathbb{P}^1 \times \mathbb{P}^1$ are

- of lower total degree and
- by nature 4-way parallel!

2-isogeny to an Extended Huff Curve

Let $a, c, d, r \in \mathbb{K}$ satisfy $acd(a - c^2 d) \neq 0, \quad r^2 = ad$.

$$H: \; y(1 + ax^2) = cx(1 + dy^2)$$

$$G: \; y(1 - ax^2) = \left(\frac{a - cr}{a + cr}\right) x(1 - ay^2).$$

$$\varphi: \; H \to G, \quad (x, y) \mapsto \left(\frac{x + \frac{r}{a}y}{1 + rxy}, \frac{x - \frac{r}{a}y}{1 - rxy}\right),$$

$$\hat{\varphi}: \; G \to H, \quad (x, y) \mapsto \left(\frac{x + y}{1 - axy}, \frac{x - y}{1 + axy} \cdot \frac{a}{r}\right).$$

Comparison - Sequential 4-NAF

Curve model	h	**cost per scalar bit**			**cost for 256 bit scalar**		
		(1,1)	(.8,.5)	(.8,0)	(1, 1)	(.8,.5)	(.8,0)
Huff	4	14.09	12.52	11.93	3608	3206	3055
Huff $a = d = 2$ *this work*	4	9.75	9.75	9.75	2496	2496	2496
Hessian , $a = \pm 1$	3	9.94	9.75	9.55	2546	2496	2445
Weierstrass $a = -3$	1	10.51	9.37	9.37	2690	2399	2399
Jacobi Intersection , $b = 1$	4	9.16	8.29	8.00	2344	2121	2049
Jacobi Quartic , $a = -1/2$	2	8.99	7.79	7.69	2301	1994	1970
Twisted Edwards , $a = -1$	4	8.40	7.62	7.62	2152	1950	1950

Each of (1,1), (.8,.5), and (.8,0) shows different **S**/**M** and **D**/**M** values, respectively, in parentheses.

Comparison - 4-way parallel

Curve model	h	**DBL**	**muADD**
Extended Huff, $a = d = 2$	4	$4 \times (2\mathbf{M})$	$4 \times (2\mathbf{M})$
Twisted Edwards, $a = -1$	4	$4 \times (1\mathbf{M} + 1\mathbf{S})$	$4 \times (2\mathbf{M})$

- **DBL** and **muADD** are the most frequent operations.
- Similar performance when 4-way parallel 1-NAF is used and $\mathbf{M} = \mathbf{S}$.
- Huff form is slower yet close in peformance when $w > 1$ for w-NAF. The reason: Twisted Edwards 4-way parallel full addition costs $4 \times (2\mathbf{M})$. But Huff slows down to $4 \times (3\mathbf{M})$.

Thank you :)

https://eprint.iacr.org/2017/320.pdf