# Elliptic curves, number theory and cryptography

## Week 14, Lecture 14B: Cryptographic Hashing to Curves

Aurore Guillevic

Aarhus University

Spring semester, 2022

These slides at

`https://members.loria.fr/AGuillevic/files/Enseignements/AU/lectures/lecture14B.pdf`

# Outline

Hashing to $\mathbb{F}_p$

Map-to-curve

# Materials

Galbraith's book: Section 11.4.3
https://www.math.auckland.ac.nz/~sgal018/crypto-book/main.pdf
IETF https://www.ietf.org/archive/id/draft-irtf-cfrg-hash-to-curve-14.html

# Outline

Hashing to $\mathbb{F}_p$

Map-to-curve

# Hashing to $\mathbb{Z}/p\mathbb{Z}$

Let $p$ be a prime and $\mathbb{Z}/p/ZZ$ the field of $p$ elements.
Given a message $m$ as a bitstring in $\{0,1\}^*$ (the $*$ means the length is not specified), how to hash into $\mathbb{Z}/p\mathbb{Z}$?

The output value $x \in \mathbb{Z}/p\mathbb{Z}$ should have a uniform distribution in $[0, p-1]$.

## Reduction modulo $p$

If $p$ has length $n$ bits, $p \in [2^{n-1}, 2^n - 1]$, the reduction mod $p$ has bias related to $p$.
If $s \in \{0,1\}^n$ is a $n$-bit string,

- $s$ mod $p$ is $s$ (because $s < p$ already) with proba $p/2^n$
- $s$ mod $p$ is $s - p$ (because $s \geq p$) with proba $1 - p/2^n$.

# Reduction modulo $p$

## Reduction modulo $p$: bias

If $p = \alpha 2^n$ with $\alpha$ a rational, $0.5 < \alpha < 1$, and $p \leq s < 2^n$, then
$0 \leq s - p < 2^n - p = (1-\alpha)2^n$.

- $s \in \{0,1\}^n$ is uniformly distributed
- $s \geq p$ with proba $1 - p/2^n = 1 - \alpha$, in this case $s \bmod p = s - p \in [0, (1-\alpha)2^n)$
- $s < (1-\alpha)2^n$ with proba $1 - \alpha$

$\Longrightarrow$ $s \bmod p \in [0, (1-\alpha)2^n]$ with proba $2(1-\alpha)$
and $s \bmod p \in [(1-\alpha)2^n, \alpha 2^n)$ with proba $2\alpha - 1$.

If $\alpha = 3/4$ ( $\Longleftrightarrow$ $p$ is roughly in the middle of $[2^{n-1}, 2^n]$):
$s \bmod p < p/3$ with probability $1/2$, and
$s \bmod p$ is **not** uniformly distributed.

## Solution

Expand the message $m$ in $\{0,1\}^{n+k}$ before reducing mod $p$.

# Reduction modulo $p$

For a bias $< 2^{-k}$ for some integer $k$,
expand $m$ as a bistring in $\{0, 1\}^{n+k}$ where $n$ is the bitsize of $p$
To ensure a security level $2^k$, a bias $2^{-k}$ is acceptable.

# Outline

Hashing to $\mathbb{F}_p$

Map-to-curve

# Hashing to curves: recommendations

The choice of the hashing technique depends on the form of the elliptic curve.

- The curve is in Montgomery form $By^2 = x^3 + Ax^2 + x$
  $\rightarrow$ Elligator-2
- The curve is in twisted Edwards form $ax^2 + y^2 = 1 + dx^2y^2$
  $\rightarrow$ twisted-Edwards Elligator-2
- The curve is in short Weierstrass form $y^2 = x^3 + ax + b$, and $ab \neq 0$
  $\rightarrow$ Simplified SWU
- The curve is in short Weierstrass form $y^2 = x^3 + ax + b$, and $ab = 0$
  $\rightarrow$ Simplified SWU for $ab = 0$, or general SWU

SWU: Shallue-van de Woestijne

# Hashing to Montgomery curves: Elligator 2

Daniel J. Bernstein, Mike Hamburg, Anna Krasnova, and Tanja Lange.
Elligator: elliptic-curve points indistinguishable from uniform random strings.
In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013*, pages 967–980. ACM Press, November 2013.

# Hashing to Montgomery curves: Elligator 2

Function `inv0` such that $\texttt{inv0}(x) = x^{p-2}$ so that

$$\texttt{inv0}(x) \begin{cases} = 0 \text{ if } x = 0 \\ = 1/x \text{ otherwise} \end{cases}$$

Function `sgn0` such that it returns a bit in $\{0, 1\}$:
$x \in \mathbb{F}_p$, $\texttt{sgn0}(x) = x \bmod 2$

# Hashing to Montgomery curves: Elligator 2

$$E\colon By^2 = x^3 + Ax^2 + x/\mathbb{F}_p,\ A, B \neq 0,\ (A-2)(A+2) \neq 0$$

$(A-2)(A+2)$ non-square $\implies$ points of order 4 but $\#E(\mathbb{F}_p)[2] = 2$, not 4.
Precomputed: a non-square $z \in \mathbb{F}_p$
Let $u \in \mathbb{F}_p$ a result of hashing to $\mathbb{F}_p$, we want to hash $u$ to the curve $E(\mathbb{F}_p)$

1. $x_1 = -(A/B) \cdot \mathtt{inv0}(1 + zu^2)$
2. If $x_1 = 0$, set $x_1 = -(A/B)$
3. $\tilde{x}_1 = x_1^3 + (A/B)x_1^2 + x_1/B^2$
4. $x_2 = -x_1 - (A/B)$
5. $\tilde{x}_2 = x_2^3 + (A/B)x_2^2 + x_2/B^2$
6. If $\mathtt{is\_square}(\tilde{x}_1)$, set $x = x_1$, $y = \sqrt{\tilde{x}_1}$ with $\mathrm{sgn}_0(y) = 1$
7. Else set $x = x_2$, $y = \sqrt{\tilde{x}_2}$ with $\mathrm{sgn}_0(y) = 0$
8. $s = x \cdot B$
9. $t = y \cdot B$
10. Return $(s, t)$

# Hashing to twisted Edwards curves:

$$ax^2 + y^2 = 1 + dx^2y^2$$

$a, d \neq 0$
First hash $u \in \mathbb{F}_p$ onto a Montgomery curve as before,
then map the point to twisted Edwards form.

# Hashing to short Weierstrass curves

## Simplified Shallue-van de Woestijne-Ulas method

Eric Brier, Jean-Sébastien Coron, Thomas Icart, David Madore, Hugues Randriam, and Mehdi Tibouchi.
Efficient indifferentiable hashing into ordinary elliptic curves.
In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 237–254. Springer, Heidelberg, August 2010.

# Simplified Shallue-van de Woestijne-Ulas method

$$E: y^2 = x^3 + ax + b = g(x)/\mathbb{F}_p,\ \ a, b \neq 0$$

$z \in \mathbb{F}_p$, non-square, $z \neq -1$, $g(x) - z \in \mathbb{F}_p[x]$ irreducible, $g(b/(za))$ square.

1. $v_1 = \texttt{inv0}(z^2 \cdot u^4 + z \cdot u^2)$
2. $x_1 = (-b/a) \cdot (1 + v_1)$
3. If $v_1 = 0$, set $x_1 = b/(z \cdot a)$
4. $\tilde{x}_1 = x_1^3 + a \cdot x_1 + b$
5. $x_2 = z \cdot u^2 \cdot x_1$
6. $\tilde{x}_2 = x_2^3 + a \cdot x_2 + b$
7. If $\texttt{is\_square}(\tilde{x}_1)$, set $x = x_1$ and $y = \sqrt{\tilde{x}_1}$
8. Else set $x = x_2$ and $y = \sqrt{\tilde{x}_2}$
9. If $\texttt{sgn0}(u) \neq \texttt{sgn0}(y)$, set $y = -y$
10. Return $(x, y)$

# Hashing to special short Weierstrass curves

$$y^2 = g(x) = x^3 + ax + b, \;\; a = 0 \text{ or } b = 0$$

Wahby–Boneh Idea: hash to an isogenous curve with $a'b' \neq 0$

Riad S. Wahby and Dan Boneh.
Fast and simple constant-time hashing to the BLS12-381 elliptic curve.
*IACR TCHES*, 2019(4):154–179, 2019.