KU LEUVEN

How to use Koblitz curves on small devices?

Kimmo Järvinen[1,2] Ingrid Verbauwhede[1]

[1] KU Leuven (Belgium)
[2] Aalto University (Finland)

CARDIS 2014, Paris, France, Nov. 5–7, 2014

- Elliptic curves are good for lightweight public-key crypto
- Koblitz curves allow very fast kP
  $\Rightarrow$ Point doublings are replaced by cheap Frobenius maps
  $\Rightarrow$ The scalar k is needed as a τ-adic expansion K
  $\Rightarrow$ **Conversions are needed and they are expensive**

- Elliptic curves are good for lightweight public-key crypto
- Koblitz curves allow very fast kP
  ⇒ Point doublings are replaced by cheap Frobenius maps
  ⇒ The scalar k is needed as a τ-adic expansion K
  ⇒ **Conversions are needed and they are expensive**
- We provide a solution to this problem:
  Conversions can be delegated to a more powerful party if the weaker party computes all operations in the τ-adic domain

Koblitz Curves

- Elliptic curves over $GF(2^m)$ of the form:

$$E : x^2 + xy = y^3 + ax^2 + 1, \text{ where } a \in \{0, 1\}$$

- If $\mathbf{P} = (x, y) \in E$, then also $F(\mathbf{P}) = (x^2, y^2) \in E$
- $2\mathbf{P} = \mu F(\mathbf{P}) - F(F(\mathbf{P}))$ where $\mu = (-1)^{1-a}$

Koblitz Curves

- Elliptic curves over $GF(2^m)$ of the form:

$$E : x^2 + xy = y^3 + ax^2 + 1, \text{ where } a \in \{0, 1\}$$

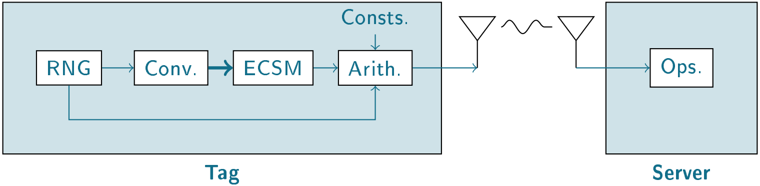
- If $\mathbf{P} = (x, y) \in E$, then also $F(\mathbf{P}) = (x^2, y^2) \in E$
- $2\mathbf{P} = \mu F(\mathbf{P}) - F(F(\mathbf{P}))$ where $\mu = (-1)^{1-a}$
- Frobenius can be seen as a multiplication by the complex number: $\tau = (\mu + \sqrt{-7})/2$
- If k is given in base-τ as $K = \sum K_i \tau^i$, then **Frobenius maps can be used** for computing $k\mathbf{P}$
  $\Rightarrow$ **Fast Frobenius-and-add** instead of slow double-and-add

Example: ECDSA Signature Generation

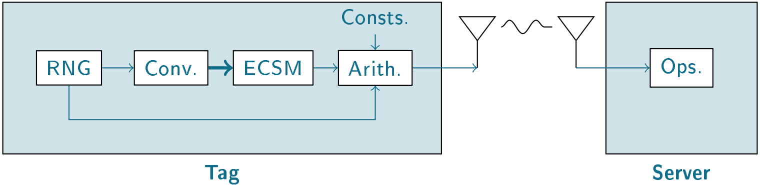
- Signature (r, s) for a message m:

$$k \in_R [1, q-1]$$
$$r = [k\mathbf{P}]_x$$
$$e = H(m)$$
$$s = k^{-1}(e + dr) \bmod q$$

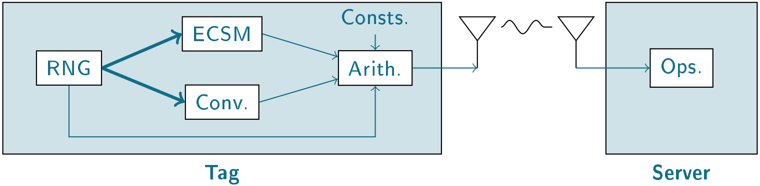
Option A: Convert a random integer to the τ-adic domain

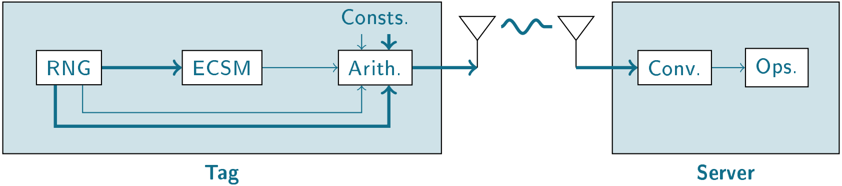


Option A: Convert a random integer to the τ-adic domain



Option B: Convert a random τ-adic expansion to an integer





- **The tag computes everything in the τ-adic domain** (values that don't depend on k can be computed normally)
- Resources are saved if **operations in the τ-adic domain are cheap** (cheaper than conversions)
- We need **an efficient algorithm for addition** of two τ-adic expansions; other arithmetic operations can be implemented using it

$a = 19 = \langle 1, 0, 0, 1, 1 \rangle$ and $b = 17 = \langle 1, 0, 0, 0, 1 \rangle$
Then, $c = 36 = \langle 1, 0, 0, 1, 0, 0 \rangle$ is given as follows:

$$
\begin{array}{ccccccc}
 & 1 & 0 & 0 & 1 & 1 & 0 \\
\hline
 & & 1 & 0 & 0 & 1 & 1 \\
+ & & 1 & 0 & 0 & 0 & 1 \\
\hline
 & 1 & 2 & 0 & 1 & 2 & 2 \\
\hline
= & 1 & 0 & 0 & 1 & 0 & 0
\end{array}
$$

- With any base $B \in \mathbb{Z}_+$, we can do:
  $r_i = a_i + b_i + t_{i-1}$
  $c_i = r_i \bmod B$
  $t_i = (r_i - c_i)/B$

KU LEUVEN

- The carry is a τ-adic number $t \in \mathbb{Z}[\tau]$ and it is uniquely given by $t = t_0 + t_1\tau$ with $t_{0,1} \in \mathbb{Z}$ (Solinas, 2000)

KU LEUVEN

- The carry is a τ-adic number $t \in \mathbb{Z}[\tau]$ and it is uniquely given by $t = t_0 + t_1\tau$ with $t_{0,1} \in \mathbb{Z}$ (Solinas, 2000)
- r_i and C_i are given similarly (only t_0 affects C_i)

$$r_i = A_i + B_i + t_0$$
$$C_i = r_i \bmod 2$$

- The carry is a τ-adic number $t \in \mathbb{Z}[\tau]$ and it is uniquely given by $t = t_0 + t_1\tau$ with $t_{0,1} \in \mathbb{Z}$ (Solinas, 2000)
- r_i and C_i are given similarly (only t_0 affects C_i)
- Division $(t - C_i)/\tau$ is given by $(t_0, t_1) \leftarrow (t_1 + \mu(t_0 - C_i)/2, -(t_0 - C_i)/2)$ (Solinas, 2000)

$$
\begin{aligned}
r_i &= A_i + B_i + t_0 \\
C_i &= r_i \bmod 2 \\
t_0 &= t_1 + \mu(t_0 - C_i)/2 \\
t_1 &= -(t_0 - C_i)/2
\end{aligned}
$$

KU LEUVEN

$A = 1 + \tau + \tau^4 = \langle 1, 0, 0, 1, 1 \rangle$
$B = 1 + \tau^4 = \langle 1, 0, 0, 0, 1 \rangle$

	1	0	0	0	1
+	1	0	0	1	1
=					

$$\begin{aligned} r_i &= t_0 + A_i + B_i \\ C_i &= r_i \bmod 2 \\ t_0 &= t_1 + \mu(t_0 - C_i)/2 \\ t_1 &= -(t_0 - C_i)/2 \end{aligned}$$

Example: τ-adic Addition ($\mu = 1$)

$A = 1 + \tau + \tau^4 = \langle 1, 0, 0, 1, 1 \rangle$
$B = 1 + \tau^4 = \langle 1, 0, 0, 0, 1 \rangle$

			-1	1	
				0	0
	1	0	0	0	1
+	1	0	0	1	1
=					0

$$r_i = 0 + 1 + 1 = 2$$
$$C_i = 2 \bmod 2 = 0$$
$$t_0 = 0 + 1 \cdot (2 - 0)/2 = 1$$
$$t_1 = -1 \cdot (2 - 0)/2 = -1$$

$A = 1 + \tau + \tau^4 = \langle 1, 0, 0, 1, 1 \rangle$
$B = 1 + \tau^4 = \langle 1, 0, 0, 0, 1 \rangle$

		-1	0		
			-1	1	
	1	0	0	0	1
+	1	0	0	1	1
=				0	0

$r_i = 1 + 0 + 1 = 2$
$C_i = 2 \bmod 2 = 0$
$t_0 = -1 + 1 \cdot (2 - 0)/2 = 0$
$t_1 = -1 \cdot (2 - 0)/2 = -1$

$A = 1 + \tau + \tau^4 = \langle 1, 0, 0, 1, 1 \rangle$
$B = 1 + \tau^4 = \langle 1, 0, 0, 0, 1 \rangle$

		0	-1				
			-1	0			
		1	0	0	0	1	
+		1	0	0	1	1	
=				0	0	0	

$r_i = 0 + 0 + 0 = 2$
$C_i = 0 \bmod 2 = 0$
$t_0 = -1 + 1 \cdot (0 - 0)/2 = -1$
$t_1 = -1 \cdot (0 - 0)/2 = 0$

Example: τ-adic Addition ($\mu = 1$)

$A = 1 + \tau + \tau^4 = \langle 1, 0, 0, 1, 1 \rangle$
$B = 1 + \tau^4 = \langle 1, 0, 0, 0, 1 \rangle$

	1	-1					
		0	-1				
		1	0	0	0	1	
+		1	0	0	1	1	
=			1	0	0	0	

$$r_i = -1 + 0 + 0 = -1$$
$$C_i = -1 \bmod 2 = 1$$
$$t_0 = 0 + 1 \cdot (-1 - 1)/2 = -1$$
$$t_1 = -1 \cdot (-1 - 1)/2 = 1$$

KU LEUVEN

$A = 1 + \tau + \tau^4 = \langle 1, 0, 0, 1, 1 \rangle$
$B = 1 + \tau^4 = \langle 1, 0, 0, 0, 1 \rangle$

	0	1					
		1	-1				
			1	0	0	0	1
+			1	0	0	1	1
=			1	1	0	0	0

$$r_i = -1 + 1 + 1 = 1$$
$$C_i = 1 \bmod 2 = 1$$
$$t_0 = 1 + 1 \cdot (1 - 1)/2 = 1$$
$$t_1 = -1 \cdot (1 - 1)/2 = 0$$

$A = 1 + \tau + \tau^4 = \langle 1, 0, 0, 1, 1 \rangle$
$B = 1 + \tau^4 = \langle 1, 0, 0, 0, 1 \rangle$

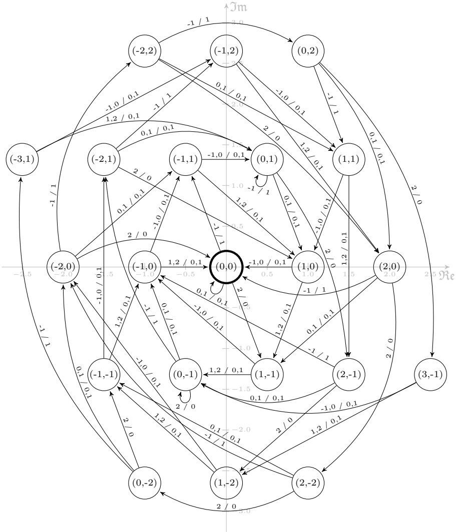
	0	0					
		0	1				
				1	0	0	0	1
+				1	0	0	1	1
=			1	1	1	0	0	0

$r_i = 1 + 0 + 0 = 1$
$C_i = 1 \bmod 2 = 1$
$t_0 = 0 + 1 \cdot (1 - 1)/2 = \mathbf{0}$
$t_1 = -1 \cdot (1 - 1)/2 = \mathbf{0}$

Hence, $C = A + B = \langle 1, 1, 1, 0, 0, 0 \rangle = \tau^3 + \tau^4 + \tau^5$

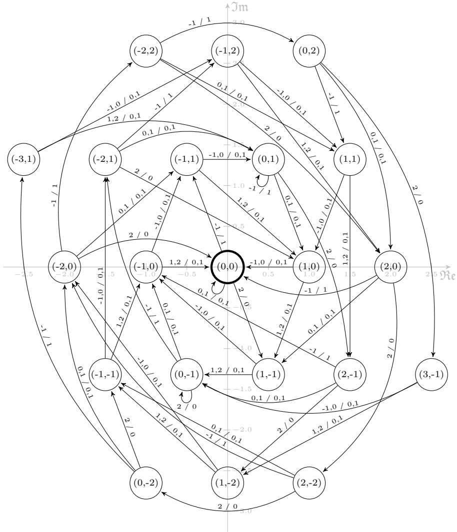
Addition Described as an FSM

- $A_i \in \{0, 1\}$ and $B_i \in \{0, \pm 1\}$ to support τNAF
- FSM includes 21 states



$\mu = 1$

- $A_i \in \{0, 1\}$ and $B_i \in \{0, \pm 1\}$ to support τNAF
- FSM includes 21 states
- The state (t_0, t_1) with $t_0 \in [-3, 3]$ and $t_1 \in [-2, 2]$
- At most 7 steps to reach $(0, 0)$ when all $A_i = B_i = 0$



$\mu = 1$

KU LEUVEN

Input: τ-adic expansions $A = \sum_{i=0}^{n-1} A_i \tau^i$ and $B = \sum_{i=0}^{n-1} B_i \tau^i$
Output: τ-adic expansion $C = A + B$ with $C_i \in \{0, 1\}$
$(t_0, t_1) \leftarrow (0, 0)$
for $i = 0$ **to** $n + 6$ **do**
  $r \leftarrow A_i + B_i + t_0$
  $C_i \leftarrow r_0$
  $(t_0, t_1) \leftarrow (t_1 + \mu \lfloor r/2 \rfloor, - \lfloor r/2 \rfloor)$

Multiplication

- shift-and-add (both operands τ-adic expansion)
- double-and-add (an integer and a τ-adic expansion)

Multiplication

- shift-and-add (both operands τ-adic expansion)
- double-and-add (an integer and a τ-adic expansion)

Inversion mod q

- Fermat's Little Theorem $A^{-1} = A^{q-2}$

Multiplication

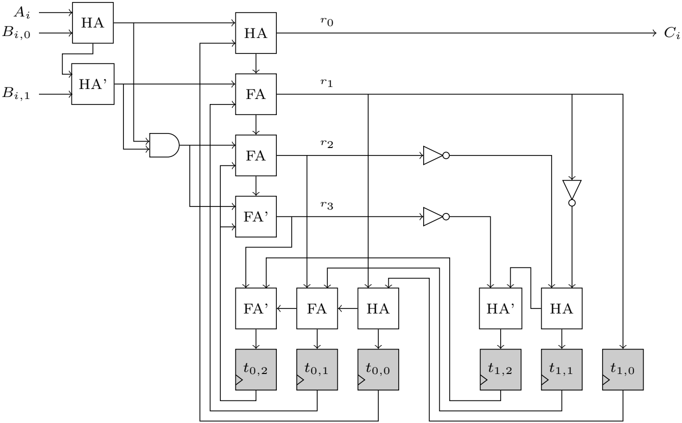
- shift-and-add (both operands τ-adic expansion)
- double-and-add (an integer and a τ-adic expansion)

Inversion mod q

- Fermat's Little Theorem $A^{-1} = A^{q-2}$

Folding

- Integer equivalent of $A = \sum_{i=0}^{n-1} A_i \tau^i$ given by $a = \sum A_i s^i \bmod q$ where s is a curve constant such that $s^m \equiv 1 \pmod{q}$ (Lange, 2005)
- Split A into m-bit blocks $A^{(0)}, \ldots, A^{(\lfloor n/m \rfloor)}$ and compute $A^{(0)} + A^{(1)} + \ldots + A^{(\lfloor n/m \rfloor)}$ with the addition algorithm
- Length of A can be reduced to approx. m


A_i
$B_{i,0}$
$B_{i,1}$
HA
HA'
HA
FA
FA
FA'
r_0
r_1
r_2
r_3
C_i
FA'
FA
HA
HA'
HA
$t_{0,2}$
$t_{0,1}$
$t_{0,0}$
$t_{1,2}$
$t_{1,1}$
$t_{1,0}$

- 130 nm CMOS, Synopsys Design Compiler, VHDL
- 75.25 GE ($\mu = 1$) or 76.25 GE ($\mu = -1$)

Work	**Technology**	**GE**
(Brumley, 2010), integer-to-τNAF	FPGA, Stratix II S60C4	$>$7200
(Brumley, 2010), τ-adic-to-integer	FPGA, Stratix II S60C4	$>$3600
This work, $\mu = 1$	ASIC, 0.13 μm CMOS	75.25
This work, $\mu = 1$	ASIC, 0.13 μm CMOS	$\sim$2000

Conclusions

- Expensive conversions can be delegated to a more powerful party by using cheap τ-adic arithmetic
- Koblitz curves are viable also for lightweight implementations

Future Work

- Side-channel countermeasures
- Bit-serial $\longrightarrow$ digit-serial
- Entire elliptic curve cryptosystem (e.g., ECDSA signing)

Thank you! Questions?