JSIAM Letters Vol.6 (2014) pp.69–72 ©2014 Japan Society for Industrial and Applied Mathematics JSIAM Letters

Computing fixed argument pairings with the elliptic net algorithm

Yang Liu[1], Naoki Kanayama[1], Kazutaka Saito[2], Tadanori Teruya[3], Shigenori Uchiyama[4] and Eiji Okamoto[1]

[1] University of Tsukuba, 1-1-1 Ten-nohdai, Tsukuba, Ibaraki 305-8573, Japan
[2] Internet Initiative Japan Inc., Jinbocho Mitsui Bldg., 1-105 Kanda Jinbo-cho, Chiyoda-ku, Tokyo 101-0051, Japan
[3] National Institute of Advanced Industrial Science and Technology, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan
[4] Tokyo Metropolitan University, 1-1 Minami-Ohsawa, Hachioji, Tokyo 192-0397, Japan

E-mail *kanayama@risk.tsukuba.ac.jp*

Received March 25, 2014, Accepted June 17, 2014

Abstract

The pairing-based cryptosystem was proposed in 2001, and it provides efficient implementations of identity-based encryption (IBE) and attribute-based encryption (ABE). In 2010, Costello and Stebila introduced the concept of fixed argument pairing, which can be applied to many applications of pairings, and, to compute these pairings, they proposed an efficient algorithm based on the Miller algorithm. In this paper, we propose a method for computing fixed argument pairings, based on the elliptic net method proposed by Stange.

Keywords pairing-based cryptography, fixed argument pairing, elliptic net algorithm

Research Activity Group Algorithmic Number Theory and Its Applications

1. Introduction

Boneh and Franklin proposed ID-based encryption based on cryptographic pairings in 2001 [1], which made the practical value of cryptographic pairings clear. Ever since pairing-based cryptosystems have been a heated topic in cryptographic research. Though acceleration of pairing computation has been widely researched so far, it still remains an important field of interest. Currently, the pairing that results in the most efficient implementation of pairing-based cryptographic schemes is the Tate pairing and some of its variants, for example, the η_T [2], Ate [3], Ate$_i$ [4], R-Ate [5], and optimal [6] pairings.

A standard algorithm for computing pairings is Miller's algorithm [7, 8]. A generic implementation of Miller's algorithm uses the classic double-and-add line-and-tangent method. Most improvements of the computation of pairings attempt to shorten the number of iterations of the so-called Miller loop in the Miller algorithm.

Fixed argument pairing is a pairing system in which one argument is fixed while the other is allowed to change. Note that though there are chances that the second argument is fixed, in this paper we discuss the situation where the first one is fixed.

Such pairings can be applied to many scenarios. For example, when we use Boneh-Franklin ID-based cryptosystem, in the decryption phase, the decryption key d_{ID} can be fixed as the sender has only one ID. The decryption key just serves as the first argument in pairing computation, so the receiver will have to compute a fixed argument pairing.

In 2010, Costello and Stebila [9] proposed a scheme based on the Miller algorithm that can be used to compute such pairings. To compute the Miller function $f_{m,P}()$, where m is an integer, they used two sets G_{DBL} and G_{ADD} to store the precomputed values. They also adopted a strategy similar to parallel computation (performing n iterations simultaneously). With P fixed, it is less expensive to compute $f_{m,P}(Q)$. According to Costello and Stebila's result, the Miller loop can be computed with between 25% to 37% fewer field operations.

In 2007, Stange [10] defined elliptic nets and proposed an alternative method based on elliptic nets for computing Tate pairings. Ogura et al. [11] presented formulas that use elliptic nets to compute cryptographic pairings.

However, using elliptic nets for pairing computations would not be very popular because for the time being the Double and DoubleAdd algorithms [10] with elliptic nets are more expensive than the classic double-and-add line-and-tangent method in Miller's algorithm.

In this paper, we propose an algorithm that uses elliptic nets to compute fixed argument pairings and show that this is an example in which a pairing computation using an elliptic net becomes more efficient. In [10], Stange considered a block $V(i)$ for an integer i, called *a block centered on* i, that consisted of 11 elliptic net values with respect to A and B to be used for computing the Tate pairing $e_r(A, B)$. Given the block $V(i)$, we can compute the block $V(2i)$ or $V(2i+1)$ by using the Double/DoubleAdd algorithm in [10]. Hence, we can compute the block $V(r)$ in polynomial time. Each block contains eight values that depend on A *only* and three

that depend on A and B. Therefore, if we precompute and store a subblock that depends only on A, then we can compute the (A, B)-dependent part by referring the table. In this way, we can efficiently compute fixed argument pairings $e(A, *)$.

This strategy is efficient when we use pairings defined on $\mathbb{G}_2 \times \mathbb{G}_1$, or roughly when $\mathbb{G}_2$ (resp. $\mathbb{G}_1$) is a cyclic group of $\mathbb{F}_{q^k}$ (resp. $\mathbb{F}_q$)-rational points on an elliptic curve $E/\mathbb{F}_q$. In this case, in $\mathbb{F}_{q^k}$ there are eight elliptic net values that depend only on A, and we can reduce the total cost of computing the fixed argument pairings. Many cryptographic pairings are defined on $\mathbb{G}_2 \times \mathbb{G}_1$ and are used widely in the implementation of pairing-based cryptosystems. In this paper, we mainly consider the case of Ate pairing.

The rest of this paper is organized as follows. In Section 2, we briefly introduce the basic principles of the Miller algorithm and the elliptic net algorithm, on which our scheme is based. In Section 3, we present the principles and details of our scheme. In Section 4, we evaluate the efficiency of the speed and storage cost of our scheme.

2. Preliminaries

2.1 Pairings

Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$ with q elements. The set of $\mathbb{F}_q$-rational points of E is denoted by $E(\mathbb{F}_q)$, and the point at infinity on E is denoted by O. Consider a large prime r such that $r \mid \#E(\mathbb{F}_q)$, and let $E(\mathbb{F}_q)[r]$ denote the subgroup of r-torsion points in $E(\mathbb{F}_q)$. The embedding degree k is the smallest positive integer such that r divides $q^k - 1$.

Let π_q be the Frobenius endomorphism $\pi_q : E \to E$, $(x, y) \mapsto (x^q, y^q)$. We denote the trace of Frobenius by t, i.e., $\#E(\mathbb{F}_q) = q + 1 - t$.

2.1.1 Tate Pairing:

Let $P \in E(\mathbb{F}_{q^k})[r]$ and $Q \in E(\mathbb{F}_{q^k})$. Choose a point $R \in E(\mathbb{F}_{q^k})$ such that the support of $\mathrm{div}(f_{r,P}) = r(P) - r(O)$ and $D_Q := (Q + R) - (R)$ are disjoint. Then, the Tate pairing is defined by

$$\langle \cdot, \cdot \rangle_r : E(\mathbb{F}_{q^k})[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) \to \mathbb{F}_{q^k}^{\times}/(\mathbb{F}_{q^k}^{\times})^r,$$

$$(P, Q) \mapsto \langle P, Q \rangle_r := f_{r,P}(D_Q) \mod (\mathbb{F}_{q^k}^{\times})^r.$$

It has been shown that $\langle P, Q \rangle_r$ is bilinear and nondegenerate.

2.1.2 Ate Pairing:

The Ate pairing, proposed by Hess et al. [3], is a generalization of the η_T pairing [2]. The Ate pairing can be applied to not only supersingular elliptic curves but also to ordinary ones.

For cryptographic applications, it is usually assumed that the points P and Q are, respectively, elements in the following groups:

$$\mathbb{G}_1 = E(\mathbb{F}_q)[r] = E(\mathbb{F}_{q^k})[r] \cap \mathrm{Ker}(\pi_q - 1),$$

$$\mathbb{G}_2 = E(\mathbb{F}_{q^k})[r] \cap \mathrm{Ker}(\pi_q - q).$$

Let $T = t - 1$. We choose integers N and L such that $N = \gcd(T^k - 1, q^k - 1)$ and $T^k - 1 = LN$. We assume that r^2 does not divide $q^k - 1$. Then the Ate pairing is defined by $f_{T,Q}^{\mathrm{norm}}(P)$ for $Q \in \mathbb{G}_2$ and $P \in \mathbb{G}_1$. Here, $f_{T,Q}^{\mathrm{norm}}$ is the normalization of $f_{T,Q}$. We denote by $\alpha(Q, P)$ the reduced Ate pairing: $\alpha(Q, P) := f_{T,Q}(P)^{(q^k-1)/r}$. The length of the Miller loop for computing the Ate pairing $f_{T,Q}(P)$ is $\log_2 |T|$.

Remark 1 *Note that the Ate paring is a "point-evaluation" pairing, although the Tate pairing $f_{r,P}(D_Q)$ is a "divisor-evaluation" pairing. The rational function with* $\mathrm{div}(f_{T,Q}) = T(Q) - ([T]Q) - (T - 1)(O)$ *is determined uniquely up to a constant. When the point Q is in $E(\mathbb{F}_{q^k})$, the constant is in $\mathbb{F}_{q^k}$, and it will not vanish during the final exponentiation. We therefore need to normalize the function. We can obtain the normalization function by $f_{T,Q}^{\mathrm{norm}} = f_{T,Q}/(z^{T-1} f_{T,Q})(O)$, where z is called the uniformizer of E on O.*

2.2 Elliptic net algorithm

At this point, strategies for accelerating the Miller algorithm seem to have been exhausted, so it is necessary to seek other methods for computing pairings. At the conference Pairing 2007, Stange defined an elliptic net and proposed a method for using it to compute a Tate pairing.

An elliptic net W is a map from a finitely generated free Abelian group $\mathcal{A}$ to an integral domain $\mathcal{R}$, that satisfies a certain recursive equation (see [10]). Stange presented the elliptic net $W_{E,P_1,P_2,\ldots,P_n}$ associated with an elliptic curve E/K, where K is a field, and $P_1, P_2, \ldots, P_n$ are points. The elliptic net $W_{E,P_1,P_2,\ldots,P_n}$ gives a function from $\mathbb{Z}^n$ to K. In this paper, we consider only the case where $n = 2$.

Suppose there is an elliptic curve defined over a finite field $\mathbb{F}_q$ with the Weierstrass function:

$$E/\mathbb{F}_q : y^2 = x^3 + Ax + B,$$

and $P = (x_1, y_1), Q = (x_2, y_2) \in E$. Then we can make an elliptic net system consisting of $W_{P,Q}(i, 0)$ and $W_{P,Q}(j, 1)$. For the initial values and recursive equations to compute the elliptic net system, see [10]. For ease of reading, here we simplify by writing $W(i, j)$ instead of $W_{P,Q}(i, j)$.

Stange also demonstrated an efficient method for computing elliptic nets. Consider a block consisting of the two vectors; $W_{P,Q}(i_1, 0)$ with $k - 3 \le i_1 \le k + 4$ and $W_{P,Q}(i_2, 1)$ with $k - 1 \le i_2 \le k + 1$. This is called an "elliptic net block centered on k". We will use "V" to denote such a block.

Stange gave two algorithms for calculating elliptic net blocks, as follows.

- Double(V): outputs an elliptic net block centered on $2k$ given the one centered on k.
- DoubleAdd(V): outputs an elliptic net block centered on $2k + 1$ given the one centered on k.

With the Double and DoubleAdd algorithms, it is possible to compute the elliptic net block centered on k given the one centered on 1 in a polynomial time of $O(\log k)$. For details, see [10].

2.3 Cryptographic pairings using elliptic nets

Stange provided the formula for using an elliptic net to find the Tate pairing.

2.3.1 Using an elliptic net to find a Tate pairing:

Theorem 2 ([10, Corollary 1]) *Let E be an elliptic curve over a finite field K. For $P \in E(K)[r]$, $Q \in E(K)$,*

$$f_{r,P}(D_Q) = \frac{W_{P,Q}(r+1,1)W_{P,Q}(1,0)}{W_{P,Q}(r+1,0)W_{P,Q}(1,1)}. \quad (1)$$

Stange also gave an algorithm for using an elliptic net to compute the Tate pairing.

2.3.2 Using an elliptic net to find an Ate pairing:

Ogura et al. [11] provided formulas for using elliptic nets to find cryptographic pairings. In this paper, we review only the formula for the Ate pairing.

Before presenting this, we first need to formulate the normalization for elliptic nets.

Proposition 3 ([11]) *$\tilde{W}_{P,Q}(s,1)$ denotes the normalization for the elliptic net $W_{P,Q}(s,1)$. For $s \in \mathbb{Z}$, assume $[s]P \neq O$. Then*

$$\tilde{W}_{P,Q}(s,1) = \frac{W_{P,Q}(s,1)}{2^{s-1}W_{P,Q}(s,0)}.$$

For practical uses of pairings, we can assume $k > 1$. In this case, $2^{(q^k-1)/r} = 1$, and so we have

$$\tilde{W}_{P,Q}(s,1)^{\frac{q^k-1}{r}} = \left(\frac{W_{P,Q}(s,1)}{W_{P,Q}(s,0)}\right)^{\frac{q^k-1}{r}}.$$

Theorem 4 ([11, Theorem 4]) *Let E be an elliptic curve over a finite field $\mathbb{F}_q$, and let $\pi_q : (x,y) \mapsto (x^q, y^q)$ be the q-Frobenius endomorphism on E. We assume that the embedding degree $k > 1$. Let r be a large prime number with $r \mid \#E(\mathbb{F}_q)$ and $(r,q) = 1$, and let $T \equiv q \pmod r$. For $P \in \mathbb{G}_1 = E(\mathbb{F}_{q^k})[r] \cap \mathrm{Ker}(\pi_q - 1)$ and $Q \in \mathbb{G}_2 = E(\mathbb{F}_{q^k})[r] \cap \mathrm{Ker}(\pi_q - q)$,*

$$\alpha(Q,P) = f_{T,Q}^{\mathrm{norm}}(P)^{\frac{q^k-1}{r}} = \tilde{W}_{Q,P}(T,1)^{\frac{q^k-1}{r}}.$$

3. Our scheme

3.1 P-dependent and (P,Q)-dependent vectors

The computations in Stange's Double/DoubleAdd algorithm are based on the recursive formulas for elliptic nets. An elliptic block can be divided into two parts $W_{P,Q}(i,0)$ and $W_{P,Q}(j,1)$. Although these are computed simultaneously in the elliptic net method, we found that $W_{P,Q}(i,0)$ can be computed independently of $W_{P,Q}(j,1)$. Consider the initial values and recursive functions above, and note that the computation of $W_{P,Q}(i,0)$ has nothing to do with $Q(x_2,y_2)$; we say that $W_{P,Q}(i,0)$ is a P-dependent vector. However, the computation of $W_{P,Q}(j,1)$ requires the values of not only $Q(x_2,y_2)$, but also $W_{P,Q}(i,0)$ and $P(x_1,y_1)$, so we call $W_{P,Q}(j,1)$ a (P,Q)-dependent vector. Note that here P-dependent and (P,Q)-dependent correspond to the standard expression of the Miller function: $f_{m,P}()$.

With this advantage of elliptic net blocks, we can easily formulate an efficient algorithm for the computation of a fixed argument pairing, in which P is fixed. When P is fixed, we can first carry out the P-dependent operations, which we will call precomputing. After obtaining these values, we can perform the (P,Q)-dependent computations, which we will call computing, at a very low cost. Our scheme includes two parts:

- Precomputing: output the values associated with the fixed argument.
- Computing: output the values associated with the changing argument. This requires the values computed in precomputing.

Table 1. Operation Counts in Each Loop.

Types	Precomputing	Computing
$\mathbb{G}_1 \times \mathbb{G}_2$	$6S_{\mathbb{F}_q} + 26M_{\mathbb{F}_q}$	$1S_{\mathbb{F}_{q^k}} + 1M_{\mathbb{F}_{q^k}} + 8M_{\mathbb{F}_{q^k}*\mathbb{F}_q}$
$\mathbb{G}_2 \times \mathbb{G}_1$	$6S_{\mathbb{F}_{q^k}} + 26M_{\mathbb{F}_{q^k}}$	$1S_{\mathbb{F}_{q^k}} + 9M_{\mathbb{F}_{q^k}}$

Our algorithms are presented in Algorithm 1 and Algorithm 2. Note that in the expression of the Miller function $f_{m,P}()$, it is important to fix m to ensure that the same computation (Double or DoubleAdd) is performed in each loop. Since m is fixed, we can use a matrix STR[t][6], in which $t = bitlength(m) - 1$, to store the precomputed values of each loop; this separates the computation of $W_{P,Q}(i,0)$ from the elliptic net method.

Algorithm 1 Precomputing

Input: $a = W_{P,Q}(2,0), b = W_{P,Q}(3,0), c = W_{P,Q}(4,0)$, $A = W_{P,Q}(2,0)^{-1}$, $m = (d_t d_{t-1} d_{t-2} d_{t-3} \dots d_0)_2$

Output: Matrix $STR[t][6]$ for storing precomputed data, $W_{P,Q}(m,0)$

$V_0 = [-a, -1, 0, 1, a, b, c, a^3c - b^3]$
for $p = t-1$ down to 0 **do**
  $S = [0,0,0,0,0,0]$
  $P = [0,0,0,0,0,0]$
  for $i = 0$ to 5 **do**
    $S[i] = V_0[i+1]^2$
    $P[i] = V_0[i]V_0[i+2]$
  end for
  if $d_p == 0$ **then**
    for $i = 1$ to 4 **do**
      $V_0[2i-2] = S[i-1] * P[i] - S[i] * P[i-1]$
      $V_0[2i-1] = (S[i-1] * P[i+1] - S[i+1] * P[i-1])A$
    end for
    for $j = 0$ to 2 **do**
      $STR[p][2j] = S[j+1]$
      $STR[p][2j+1] = P[j+1]$
    end for
  else
    for $i = 1$ to 4 **do**
      $V_0[2i-2] = (S[i-1] * P[i+1] - S[i+1] * P[i-1])A$
      $V_0[2i-1] = S[i] * P[i+1] - S[i+1] * P[i]$
    end for
    for $j = 0$ to 2 **do**
      $STR[p][2j] = S[j+2]$
      $STR[p][2j+1] = P[j+2]$
    end for
  end if
end for
return $STR, V_0[3](W_{P,Q}(m,0))$

Algorithm 2 Computing

Input: $d = W_{P,Q}(2,1)$, $g = W_{P,Q}(1,1)$, $E = W_{P,Q}(-1,1)^{-1}$, $F = W_{P,Q}(2,-1)^{-1}$, $G = W_{P,Q}(1,1)^{-1}$, $m = (d_t d_{t-1} d_{t-2} d_{t-3} \dots d_0)_2$, precomputed data $STR[t][6]$

Output: $W_{P,Q}(m,1)$

1: $V_1 = [1, g, d]$
2: **for** $p = t-1$ down to 0 **do**
3: $S_0 = V_1[1]^2$
4: $P_0 = V_1[0]V_1[2]$
5: **if** $d_p == 0$ **then**
6: $V_1[0] = (P_0 * STR[p][0] - STR[p][1] * S_0)G$
7: $V_1[1] = STR[p][2] * P_0 - S_0 * STR[p][3]$
8: $V_1[2] = (STR[p][4] * P_0 - S_0 * STR[p][5])E$
9: **else**
10: $V_1[0] = STR[p][0] * P_0 - S_0 * STR[p][1]$
11: $V_1[1] = (STR[p][2] * P_0 - S_0 * STR[p][3])E$
12: $V_1[2] = (S_0 * STR[p][5] - STR[p][4] * P_0)F$
13: **end if**
14: **end for**
15: **return** $V_1[1]$ $(W_{P,Q}(m,1))$

4. Evaluation of the proposed method

4.1 Computational cost

There are different types of computation in Precomputing and Computing phase. The types of computation in different kinds of pairings are not the same either. We begin by listing the operation counts of precomputing and computing, in Table 1. The meanings of the notations used in the table are as follows:

$M_{\mathbb{F}_q}$: The multiplication of two elements in $\mathbb{F}_q$;

$M_{\mathbb{F}_{q^k}}$: The multiplication of two elements in $\mathbb{F}_{q^k}$;

$M_{\mathbb{F}_{q^k}*\mathbb{F}_q}$: The multiplication of an element in $\mathbb{F}_{q^k}$ and one in $\mathbb{F}_q$;

$S_{\mathbb{F}_q}$: The square of an element in $\mathbb{F}_q$;

$S_{\mathbb{F}_{q^k}}$: The square of an element in $\mathbb{F}_{q^k}$;

Our scheme can reduce the number of computations of the loop of the elliptic net method down to $1S_{\mathbb{F}_{q^k}} + 9M_{\mathbb{F}_{q^k}}$. The reduction is equal to the operations in precomputing. The types of operations saved depend on the type of pairings. We see that the time needed for the computing phase for a $\mathbb{G}_2 \times \mathbb{G}_1$ pairing, denoted as t_{pre}, is exactly $(1S + 9M)/(7S + 35M)t_{no_pre}$, where t_{no_pre} denotes the time needed for computing a $\mathbb{G}_2 \times \mathbb{G}_1$ pairing using the elliptic net method. When the characteristic p is neither 2 nor 3, we can take $S = 0.8M$. In this situation, $t_{pre} = (9.8/40.6)t_{no_pre} = 0.241t_{no_pre}$. However, in $\mathbb{G}_1 \times \mathbb{G}_2$ pairings, because the types of operations are different in precomputing and computing (the operations in computing require more time), the relationship between t_{no_pre} and t_{pre} is $t_{pre} > 0.241t_{no_pre}$.

4.2 Storage cost

For each $W_{P,Q}(i,0)$ vector, there are eight elements. However, since it is necessary to fix certain elements, it is only necessary to store six elements for each loop. The storage costs for $\mathbb{G}_1 \times \mathbb{G}_2$ and $\mathbb{G}_2 \times \mathbb{G}_1$ pairings are listed in Table 2.

Table 2. Storage Cost.

Types	Field	Number
$\mathbb{G}_1 \times \mathbb{G}_2$	$\mathbb{F}_q$	$6 * \log_2 m$
$\mathbb{G}_2 \times \mathbb{G}_1$	$\mathbb{F}_{q^k}$	$6 * \log_2 m$

5. Conclusion

In this paper, we proposed a method based on the elliptic net method proposed by Stange in [10], for computing fixed argument pairings. According to our analysis, our method saves about 70% of the time cost, relative to the Elliptic Net Method without precomputation, for computing $\mathbb{G}_2 \times \mathbb{G}_1$ pairings.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers 22300002, 22500005, 24540135.

References

[1] D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing, in: Proc. of CRYPTO 2001, J. Kilian ed., LNCS, Vol. 2139, pp. 213–369, Springer-Verlag, Berlin, 2001.
[2] P. S. L. M. Barreto, S. D. Galbraith, C. ÓhÉigeartaigh and M. Scott, Efficient pairing computation on supersingular abelian varieties, Des. Codes Cryptogr., **42** (2007), 239–271.
[3] F. Hess, N. P. Smart and F. Vercauteren, The Eta pairing revisited, IEEE Trans. Inform. Theory, **52** (2006), 4595–4602.
[4] C.-A. Zhao, F. Zhang and J. Huang, A note on the Ate pairing, Int. J. Inf. Secur., **6** (2008), 379–382.
[5] E. Lee, H. S. Lee and C. M. Park, Efficient and generalized pairing computation on abelian varieties, IEEE Trans. Inform. Theory, **55** (2009), 1793–1803.
[6] F. Vercauteren, Optimal pairings, IEEE Trans. Inform. Theory, **56** (2010), 455–461.
[7] V. S. Miller, Short programs for functions on curves, 1986, `http://crypto.stanford.edu/miller/miller.pdf`.
[8] V. S. Miller, The Weil pairing and its efficient calculation, J. Cryptology, **17** (2004), 235–261.
[9] C. Costello and D. Stebila, Fixed argument pairings, in: Proc. of LATINCRYPT 2010, M. Abdalla et al. eds., LNCS, Vol. 6212, pp. 92–108, Springer-Verlag, Berlin, 2010.
[10] K. E. Stange, The Tate pairing via elliptic nets, in: Proc. of Pairing 2007, T. Takagi et al. eds., LNCS, Vol. 5475, pp. 329–348, Springer-Verlag, Berlin, 2007.
[11] N. Ogura, N. Kanayama, S. Uchiyama and E. Okamoto, Cryptographic pairings based on elliptic nets, in: Proc. of IWSEC 2011, T. Iwata et al. eds., LNCS, Vol. 7038, pp. 65–78, Springer-Verlag, Berlin, 2011.