Generalised Cycling Attacks on RSA and Strong RSA Primes

Marc Gysin and Jennifer Seberry

Centre for Computer Security Research *
School of Information Technology and Computer Science
The University of Wollongong
Wollongong, NSW 2522
Australia
[marc,jennie]@cs.uow.edu.au

Abstract. Given an RSA modulus n, a ciphertext c and the encryption exponent e, one can construct the sequence

$$x_0 = c \bmod n, \quad x_{i+1} = x_i^e \bmod n, \quad i = 0, 1, \ldots$$

until $gcd(x_{i+1} - x_0, n) \neq 1$ or $i > B$, B a given boundary. If $i \leq B$, there are two cases. Case 1: $gcd(x_{i+1} - x_0, n) = n$. In this case $x_i = m$ and the secret message m can be recovered. Case 2: $1 \neq gcd(x_{i+1} - x_0, n) \neq n$. In this case, the RSA modulus n can be factorised. If $i \leq B$, then Case 2 is much more likely to occur than Case 1. This attack is called a *cycling attack*. We introduce some new *generalised cycling attacks*. These attacks work *without* the knowledge of e and c. Therefore, these attacks can be used as factorisation algorithms. We also translate these attacks to elliptic curves. For this case we call these attacks *EC generalised cycling attacks*. Finally, we review criteria that a strong RSA prime must satisfy.

1 Preliminaries

The reader is assumed to be familiar with the RSA cryptosystem, [RivShaAdl78]. A brief introduction to Lucas sequences and elliptic curves is given in the appendix. Throughout this paper we will use the following notations. If $x_0, x_1, x_2, \ldots$ is a sequence of elements, then $\{X\}$ will denote the whole sequence. If the elements are taken modulo a certain number, say p, and the sequence is periodic, then we will denote its period by $\pi_{\{X\},p}$. We write $a \mid b$ for a divides b. $(a|n)$ denotes the Legendre or Jacobi symbol if n is prime or composite, respectively.

1.1 The Carmichael and Omega Function

We will make use of the Carmichael and Omega functions $\lambda(\cdot)$ and $\Omega(\cdot,\cdot)$, respectively. $\lambda(\cdot)$ is defined as follows (see, for example, [Riesel85]):

* Supported by ARC Large Grants A9803826, A49703117

$$\lambda(2) = 1,\ \lambda(4) = 2,\ \lambda(8) = 2,$$

and for $k > 3$

$$\lambda(2^k) = 2\lambda(2^{k-1}).$$

For prime $p \geq 3$ and $k > 1$ we have

$$\lambda(p) = p - 1,\quad \lambda(p^k) = p\lambda(p^{k-1}).$$

Finally, for $n = p_1^{e_1} \cdot \ldots \cdot p_k^{e_k}$, p_i prime, $e_i \geq 1$:

$$\lambda(n) = lcm(\lambda(p_1^{e_1}), \ldots, \lambda(p_k^{e_k})).$$

The Carmichael function $\lambda(\cdot)$ and the well known Euler totient function $\phi(\cdot)$ are intimately connected. If, for example, $U(Z_n)$ denotes the multiplicative group of units in Z_n, then we can describe the following via these two functions:

- the order of the group $|U(Z_n)|$ is equal to $\phi(n)$;
- the maximum order of an element $z \in U(Z_n)$ is $\lambda(n)$;
- as a consequence of the above two statements: $U(Z_n)$ is cyclic if and only if $\lambda(n) = \phi(n)$.

The Omega function $\Omega(\cdot, \cdot)$ is defined as follows:

$$\Omega(2, D) = \begin{cases} 1 & D \text{ is even} \\ 3 & D \text{ is odd} \end{cases}$$

and for $k > 1$

$$\Omega(2^k, D) = 2\Omega(2^{k-1}, D).$$

For prime $p \geq 3$ and $k > 1$ we have

$$\begin{aligned} \Omega(p, D) &= p - (D|p),\ (D|p) \neq 0 \\ \Omega(p, D) &= 2,\ (D|p) = 0, \\ \Omega(p^k, D) &= p\Omega(p^{k-1}, D) \end{aligned}$$

Finally, for $n = p_1^{e_1} \cdot \ldots \cdot p_k^{e_k}$, p_i prime, $e_i \geq 1$:

$$\Omega(n, D) = lcm(\Omega(p_1^{e_1}, D), \ldots, \Omega(p_k^{e_k}, D)).$$

2 Generalised Cycling Attacks on RSA Moduli $n = pq$

2.1 Introduction: The Function $Enc(\cdot, \cdot)$

In this section we show that all the attacks involve the same function, subsequently called the $Enc(\cdot, \cdot)$ function. $Enc(\cdot, \cdot)$ means a generalised RSA or Rabin encryption function. In this paper we define $Enc(\cdot, \cdot)$ for three different mathematical settings, namely:

- the multiplicative group of $U(Z_n)$ or $U(Z_p)$;
- Lucas sequences $V(P,1) \bmod p$ or $V(P,1) \bmod n$;
- the additive group of points on an elliptic curve $E(F_p)$ or $E(Z_n)$.

Other mathematical settings are possible. Let m be an element of the corresponding mathematical setting $\mathcal{M}$ and let $x \in Z$. Then the function $Enc(\cdot,\cdot) : \mathcal{M} \times Z \rightarrow \mathcal{M}$ is defined as follows.

$$Enc(m,x) = \begin{cases} m^x & \text{for integers modulo a prime or composite} \\ V_x(m,1) & \text{for Lucas sequences modulo a prime or composite} \\ x \cdot m & \text{for elliptic curves over } F_p \text{ or } Z_n \end{cases}$$

Note that m must be an element of the corresponding mathematical setting $\mathcal{M}$. In particular, for the last case (elliptic curve), m must be a point on the elliptic curve. From the definition of $Enc(\cdot,\cdot)$, we have $Enc(Enc(m,x_1),x_2) = Enc(Enc(m,x_2),x_1) = Enc(m,x_1x_2)$. Let now, for prime p and $D = P^2 - 4$

$$\Psi(p) = \begin{cases} \lambda(p) & \text{for integers } \bmod p \\ \Omega(p,D) & \text{for Lucas sequences } \bmod p \\ p+1+t_1 & \text{for elliptic curves over } F_p \end{cases}$$

and for RSA modulus $n = pq$ and $D = P^2 - 4$

$$\Psi(pq) = \begin{cases} \lambda(pq) & \text{for integers } \bmod n \\ \Omega(pq,D) & \text{for Lucas sequences } \bmod n \\ (p+1+t_1)(q+1+t_2) & \text{for elliptic curves over } Z_n \end{cases}$$

where $(p+1+t_1)$ and $(q+1+t_2)$ are the orders of the additive groups of the elliptic curves over F_p and F_q, respectively. Observe that now $Enc(m,x) \equiv m \bmod p$ for $x \equiv 1 \bmod \Psi(p)$ and $Enc(m,x) \equiv m \bmod n$ for $x \equiv 1 \bmod \Psi(n)$ [1]. For the cycling attacks one now hopes that either $F(\Psi(p))$ or $F(\Psi(q))$ for this mathematical setting is smooth (smooth is defined below), that is, $F(\Psi(p))$ or $F(\Psi(q))$ has only small factors. $F(\cdot)$ is a function which depends on the particular attack chosen. For each of the three mathematical settings, there are now two possible cycling attacks.

Attack 1: $x_0 = Enc(seed, e^0)$, $x_{i+1} = Enc(x_i, e) = Enc(seed, e^{i+1})$, $i = 0, 1, \ldots$

Attack 2: $x_0 = Enc(seed, \tilde{V}_0(\tilde{P},1))$, $x_{i+1} = Enc(seed, \tilde{V}_{i+1}(\tilde{P},1))$, $i = 0, 1, \ldots$

where $\tilde{V}(\tilde{P},1)$ is a Lucas sequence.

---

[1] The reader may have noted that $\Psi(\cdot)$ does not necessarily coincide with the order/period of the corresponding mathematical setting. However, the following statement is always true: the maximum order/period of an element of the corresponding mathematical setting divides $\Psi(\cdot)$. Also to be more precise, the function $\Psi(\cdot)$ has one, two or three arguments depending on whether one is working with integers modulo a prime or composite, Lucas sequences or elliptic curves, respectively. This is omitted for the sake of simplicity.

The cycling attacks described under Attack 1 will have a complexity whose upper bound is given by $min(\lambda(\Psi(p)), \lambda(\Psi(q)))$ whereas the upper bound of the complexity of the cycling attacks described under Attack 2 will be $min(\Omega(\Psi(p), \tilde{P}^2 - 4), \Omega(\Psi(q), \tilde{P}^2 - 4))$. More precisely, the running times of attacks belonging to Attack 1 will divide $O(min(\lambda(\Psi(p)), \lambda(\Psi(q))))$ whereas the running times of attacks belonging to Attack 2 will divide $O(min(\Omega(\Psi(p), \tilde{P}^2 - 4), \Omega(\Psi(q), \tilde{P}^2 - 4)))$.

In the subsequent sections, we describe cycling attacks mod n. All the sequences $\{X\}$ have a possible empty aperiodic part and a periodic part (or cycle) of length $\pi_{\{X\},n}$. To understand the behaviour of $\{X\} = x_i \bmod n$, we must first study $\{X\} \bmod p$ and calculate the period $\pi_{\{X\},p}$, that is, the length of the periodic part of $\{X\} \bmod p$. The aperiodic part, if it exists, is usually very small ($O(\log p)$ elements). All the attacks have the following calculations in common:

Algorithm Cycling Attack:

Input: n, starting values $seed, start_0, \ldots$;
parameters $par_0, par_1, \ldots$, *a boundary B*
Output: "success",p,q; or "fail"

---

set $x_0 = Enc(seed, \cdot)$ *mod n;*
set $start = \lfloor \log n \rfloor$;
repeat
set $x_{i+1} = Enc(x_i, \cdot)$ *mod n;*
until $i \geq start$;
repeat
set $x_{i+1} = Enc(x_i, \cdot)$ *mod n;*
set $test = gcd(x_{i+1} - x_{start}, n)$;
until $test \neq 1$ *or* $i > B$;
If $test \neq 1$ *and* $test \neq n$ *then Output("success",test,n:test); else Output("fail");*

Instead of starting with *seed*, a random number, it is a good idea to let $seed = Enc(seed, 2)$. Since all the group orders/periods considered in this paper are even, x_0 may now have half of the original order, or the sequence associated with x_0 may now have half of the original period. This would increase the efficiency of the algorithm by 100%. The only purpose of the first repeat–loop is to skip a possible aperiodic part of the sequence $\{X\}$. (This repeat loop can be omitted if it is known that $\{X\}$ can not have an aperiodic part.) That is, x_{start} should now be in the periodic part of $\{X\}$. In the second repeat–loop we are trying to factorise n. (As in many factorisation algorithms we could accumulate a product of x_i's mod n and test for the *gcd* in, say, every 100th step to speed up the performance.) There are now three possible cases:

- The period $\pi_{\{X\},p}$ or $\pi_{\{X\},q}$ is less or equal than the boundary B: and $\pi_{\{X\},p} \neq \pi_{\{X\},q}$: In this case, $1 < test < n$ and the algorithm succeeds.
- The periods $\pi_{\{X\},p}$ and $\pi_{\{X\},q}$ are less or equal than the boundary B; and $\pi_{\{X\},p} = \pi_{\{X\},q}$: This case occurs with very low probability. In this case

$test = n$ and the algorithm fails. We can simply retry the algorithm with some other *seed* and/or parameters.
- The periods $\pi_{\{X\},p}$ and $\pi_{\{X\},q}$ are greater than the boundary B: In this case $test = 1$ and the algorithm fails.

All the attacks subsequently described have in common that the next element x_{i+1} in the sequence can be calculated from the current element x_i in $O(1)$ steps. We need one theorem for further discussions. A proof of this theorem is given in the technical report, [GysSeb98].

Theorem 1. *Let* $\{X\}$ *and* $\{Y\}$ *be two sequences. Let* $\{X\} = x_i = seed^i$ *and* $\{Y\} = y_i = V_i(P, 1)$, *let* $n > 0$, $gcd(seed, n) = 1$. *Then*

(i) the period $\pi_{\{X\},n}$ *of* $\{X\}$ *mod* n *satisfies:* $\pi_{\{X\},n} \mid \lambda(n)$;
(ii) the period $\pi_{\{Y\},n}$ *of* $\{Y\}$ *mod* n *satisfies:* $\pi_{\{Y\},n} \mid \Omega(n, P^2 - 4)$.

2.2 $Enc(\cdot, \cdot)$: Integers mod n

Attack 1 Let

$$x_0 = seed \bmod p, \quad x_{i+1} = x_i^e \bmod p$$

where $seed, e \geq 2$. We examine the period of $x_i \bmod p$. We first note that $x_i = seed^{e^i} \bmod p$. We consider $e^i \bmod p - 1$. [2] If $p - 1 = e^t \ell$ and $gcd(e, \ell) = 1$, then

- the above sequence will have a maximum aperiodic part of length t;
- for the period $\pi_{\{X\},p}$ we have $\pi_{\{X\},p} \mid \lambda(\ell)$.

The algorithm has a high chance of success if either $\lambda(\lambda(p))$ or $\lambda(\lambda(q))$ is either small or smooth.

Prevention of the attack:
To prevent this attack the designer of a public-key cryptosystem or CSPRBG of the RSA or Rabin type must choose p and q such that $\lambda(\lambda(p))$ and $\lambda(\lambda(q))$ are not small and not smooth. In particular, a strong prime p designed to withstand this attack must have the following properties:

- $p - 1$ must have a large factor, say t;
- $t - 1$ must have a large factor.

Similar statements can be made about the other prime q.

[2] More precisely, we have to consider e^i modulo the order of *seed* mod p. Since the order of *seed* mod p divides $p - 1$ and $\lambda(x) \mid \lambda(y)$ and $\Omega(x, D) \mid \Omega(y, D)$ for $x \mid y$ and we only state $\pi_{\{X\},p}$ divides some number, everything works out nicely at the end. The aperiodic part might be less than the maximum number stated because the order of *seed* mod p divides $p - 1$ and is not necessarily equal to $p - 1$. Similar considerations need to be made for all the other algorithms but they are omitted for the sake of simplicity.

Attack 2 Let

$$x_0 = seed^{V_0} \bmod p, \quad x_1 = seed^{V_1} \ mod p, \quad x_{i+1} = seed^{V_{i+1}} \bmod p$$

where $seed \geq 2$ and $V(P,1)$ is a Lucas sequence. We examine the period of $x_i \bmod p$. We consider $V(P,1) \bmod p-1$ and note that $\pi_{V(P,1),p-1} \mid \Omega(p-1, P^2-4)$. There is no aperiodic part. The algorithm has a high chance of success if either $\Omega(\lambda(p), P^2-4) = \Omega(p-1, P^2-4)$ or $\Omega(\lambda(q), P^2-4) = \Omega(q-1, P^2-4)$ is either small or smooth.

Remark:
In the repeat loop $x_{i+1} = x_i^P \cdot x_{i-1}^{-1} = seed^{PV_i(P,1)-V_{i-1}(P,1)} = seed^{V_{i+1}(P,1)}$. Therefore, one does not need to keep track of the individuals values of $V(P,1)$ since these are calculated implicitly.

Prevention of the attack:
To prevent this attack the designer of a public–key cryptosystem or CSPRBG of the RSA or Rabin type must choose p and q such that $\Omega(\lambda(p), D)$ and $\Omega(\lambda(q), D)$ are not small and not smooth for any values of D. In particular, a strong prime p designed to withstand this attack must have the following properties:

- $p-1$ must have a large factor, say t;
- $t-1$ *and* $t+1$ must have a large factor.

Similar statements can be made about the other prime q.

2.3 $Enc(\cdot,\cdot)$: Lucas Sequences

Attack 1 Let

$$x_0 = V_1(P,1) \bmod p, \quad x_{i+1} = V_{e^{i+1}}(P,1) \bmod p$$

where $e \geq 2$. We examine the period of $x_i \bmod p$. If $(P^2-4|p) = 1$, then we have to examine $e^j \bmod p-1$, if $(P^2-4|p) = -1$, then we have to examine $e^j \bmod p+1$. The case $(P^2-4|p) = 0$ occurs with neglible probability. The case $(P^2-4|p) = 1$ has the same complexity as the attack in Section 2.2 (since in this case $\lambda(\Omega(p, P^2-4)) = \lambda(\lambda(p))$). Therefore, we assume $(P^2-4|p) = -1$. If, now, $p+1 = e^t\ell$ and $gcd(e,\ell) = 1$, then

- the above sequence will have a maximum aperiodic part of length t;
- for the period $\pi_{\{X\},p}$ we have $\pi_{\{X\},p} \mid \lambda(\ell)$.

The algorithm has a high chance of success if either $\lambda(\Omega(p, P^2-4))$ or $\lambda(\Omega(q, P^2-4))$ is either small or smooth.

Remarks:
(i) We need to explain how to calculate $V_{e^{i+1}}(P,1) \bmod p$ from $V_{e^i}(P,1) \bmod p$. We let

$$M = \begin{bmatrix} 0 & 1 \\ -1 & P \end{bmatrix},$$

then V_j can easily be derived from M^j (see also appendix). (ii) For $e = 2$ the calculations in the repeat-loops can be simplified to $x_{i+1} = x_i^2 - 2 \bmod n$. This is because $x_{i+1} = V_2(x_i, 1) = x_i^2 - 2 \bmod n$.

Prevention of the attack:
To prevent this attack the designer of a public-key cryptosystem or CSPRBG of the RSA or Rabin type must choose p and q such that $\lambda(\Omega(p, D))$ and $\lambda(\Omega(q, D))$ are not small and not smooth for any values of D. In particular, a strong prime p designed to withstand this attack must have the following properties:

- $p - 1$ *and* $p + 1$ must have a large factor, say t and w;
- $t - 1$ *and* $w - 1$ must have a large factor.

Similar statements can be made about the other prime q.

Attack 2 Let $V(P, 1)$ and $\tilde{V}(\tilde{P}, 1)$ be two Lucas sequences. Let

$$x_0 = V_{\tilde{V}_0(\tilde{P},1)}(P, 1) \bmod p, \quad x_{i+1} = V_{\tilde{V}_{i+1}(\tilde{P},1)}(P, 1) \bmod p$$

We examine the period of $x_i \bmod p$. Since $V(P, 1) \bmod p$ has a period which divides $\Omega(p, P^2 - 4)$, $x_i \bmod p$ has a period which divides $\Omega(\Omega(p, P^2 - 4), \tilde{P}^2 - 4)$. There is no aperiodic part. Note that $\Omega(p, P^2 - 4) = p - 1$, if $\left(P^2 - 4|p\right) = 1$ and $\Omega(p, P^2 - 4) = p + 1$, if $\left(P^2 - 4|p\right) = -1$. The case $\left(P^2 - 4|p\right) = 0$ occurs with neglible probability. We only examine the case $\left(P^2 - 4|p\right) = -1$ and, if $p + 1$ has a large factor, say t, $\left(\tilde{P}^2 - 4|t\right) = -1$ since all the other cases are implicitly covered by the above attacks.

Remarks:
(i) Again we need to describe how one can calculate $V_{\tilde{V}_{i+1}(\tilde{P},1)}(P, 1)$ from $V_{\tilde{V}_i(\tilde{P},1)}(P, 1)$ in $O(1)$ steps. The idea is similar to the above. Let M be as above and let M_i be a sequence of 2×2 matrixes. In particular,

$$M_0 = M^{\tilde{V}_0(\tilde{P},1)} = M^2, \quad M_1 = M^{\tilde{V}_1(\tilde{P},1)} = M^{\tilde{P}}$$

and then

$$M_{i+1} = M_i^{\tilde{P}} \times M_{i-1}^{-1} = M^{\tilde{P}\tilde{V}_i(\tilde{P},1) - \tilde{V}_{i-1}(\tilde{P},1)} = M^{\tilde{V}_{i+1}(\tilde{P},1)}.$$

Therefore, there is no need to keep track of the individual values of $\tilde{V}(\tilde{P}, 1)$ since these are calculated implicitly. (ii) A simple implementation of this algorithm turns out to be about two to three times slower than a simple implementation of the algorithm in Section 2.2, due to the many matrix-operations involved. However, this algorithm is the most general one, in the sense that it induces the strongest requirement on a strong prime p or q (see also below). In particular, a strong prime p or q designed to withstand this attack withstands all previous attacks.

The algorithm has a high chance of success if either $\Omega(\Omega(p, P^2-4), \tilde{P}^2-4)$ or $\Omega(\Omega(q, P^2-4), \tilde{P}^2-4)$ is either small or smooth.

Prevention of the attack:
To prevent this attack the designer of a public–key cryptosystem or CSPRBG of the RSA or Rabin type must choose p and q such that $\Omega(\Omega(p, D), \tilde{D})$ and $\Omega(\Omega(q, D), \tilde{D})$ are not small and not smooth for any values of D and $\tilde{D}$. In particular, a strong prime p designed to withstand this attack must have the following properties:

- $p-1$ *and* $p+1$ must have a large factor, say t and w;
- $t-1$, $w-1$ *and* $t+1$, $w+1$ must have a large factor.

Similar statements can be made about the other prime q.

2.4 *Enc*$(\cdot,\cdot)$: Elliptic Curves

Two more attacks involving elliptic curves are elaborated in the following sections. These are slightly different to the generalised cycling attacks, since there might be a failure of the inversion step during the addition of points on the elliptic curve (which is the most welcome since then we can factorise n). However, the general idea is exactly the same except that the mathematical setting involved is the additive group of points on an elliptic curve. The elliptic curves are the most promising because of the large variety of group orders they offer. That is, if below one elliptic curve "does not work" there is a chance that another one "does work and will be successful".

Attack 1 Let $x_0 = P = (x, y)$ be a point on an elliptic curve over F_p, let $e \geq 2$. We then form the sequence of points $\{X\}$, where $x_{i+1} = e \cdot x_i$. (That is, $x_{i+1} = x_i + x_i + \ldots + x_i$, where '+' is performed e times and '+' corresponds to the addition of two points on the elliptic curve over F_p.) Let $o = \#E(F_p)$ denote the number of points on this particular elliptic curve. (We do not make any further considerations about the group structure of $E(F_p)$. This does not falsify our analyses - however upper bounds given could be slightly improved by considering such group structures.) If $o = e^t\ell$ and $gcd(e, \ell) = 1$, then

- the above sequence will have a maximum aperiodic part of length t;
- for the period $\pi_{\{X\},p}$ we have $\pi_{\{X\},p} \mid \lambda(\ell)$.

The algorithm:
The algorithm takes a point $P = (x, y)$ and the parameter a as an input. These determine the elliptic curve $y^2 = x^3 + ax + b \bmod n$ uniquely. The function $xcoord(P)$ returns the x–coordinate of the point P. There are now two possible outcomes that lead to the factorisation of n: (i) $x_{i+1} = x_{start} \bmod p$, or, $x_{i+1} = x_{start} \bmod q$ but not both (in fact, $xcoord(x_{i+1}) = xcoord(x_{start}) \bmod p$, or, $xcoord(x_{i+1}) = xcoord(x_{start}) \bmod q$ but not both is sufficient and may occurs earlier). (ii) The inversion step for the partial addition of two points on $E(Z_n)$

fails. This is indicated by the variable *invfail* in the algorithm. If this occurs, then the variable *test* will be set accordingly, that is, *test* now holds p or q.

An algorithm for factorising $n = pq$ can now be sketched as follows:

Input: n, $P = (x, y)$, a, e, B; *Output:* "success",p,q; *or* "fail"

---

set $b = y^2 - x^3 - ax \bmod n$;
set $start = \lfloor \log_e n \rfloor$;
set $x_0 = P$;
repeat
  set $x_{i+1} = e \times x_i$;
until $i \geq start$;
repeat
  set $x_{i+1} = e \times x_i$; (* *This sets also invfail and test* *)
  if not invfail then set test $= gcd(xcoord(x_{i+1}) - xcoord(x_{start}), n)$;
until test $\neq 1$ *or invfail or* $i > B$;
If test $\neq 1$ *and test* $\neq n$ *then Output("success",test,n:test); else Output("fail");*

Remarks:
(i) Instead of testing $gcd(xcoord(x_{i+1}) - xcoord(x_{start}), n)$, we could test $gcd(ycoord(x_{i+1}) - ycoord(x_{start}), n)$ or both. (ii) Doubling the x–coordinate of a point is independent of the y–coordinate. Therefore, for $e = 2$, the algorithm can be simplified as follows: choose $a, b, x_0 \in Z_n$ such that $x_0^3 + ax_0 + b$ is a square. In the repeat–loops we set:

$$x_{i+1} = \frac{x_i^4 - 2ax_i^2 + a^2 - 8x_ib}{4x_i^3 + 4ax_i + 4b} \bmod n.$$

This equation is obtained from the doubling of point equation and the elliptic curve equation and some simple transformations.

Prevention of the attack:
Since the orders o of various elliptic curves are in between $p+1-t$ and $p+1+t$, where $t^2 = 4p$, it is impossible to design a strong prime to withstand *all* of these specific attacks. The best advice is to choose a large prime p. "Large" depends on security requirements and on the amount of computing cycles that can be performed in a given time unit. This will be discussed in another paper.

Attack 2 Let $P = (x, y)$ be a point on an elliptic curve over F_p. Let $V(\tilde{P}, 1)$ be a Lucas sequence. We then form the sequence of points $\{X\}$, where $x_0 = V_0(\tilde{P}, 1) \cdot P$, $x_{i+1} = V_{i+1}(\tilde{P}, 1) \cdot P$. Let $o = \#E(F_p)$ denote the number of points on this particular elliptic curve. For the period $\pi_{\{X\},p}$ we now have $\pi_{\{X\},p} \mid \Omega(o, \tilde{P}^2 - 4)$. There is no aperiodic part.

The algorithm:
The algorithm takes a point $P = (x, y)$ and the parameter a as an input. These determine the elliptic curve $y^2 = x^3 + ax + b \bmod n$ uniquely. The function $xcoord(P)$ returns the x–coordinate of the point P. As above there are now two possible outcomes that lead to the factorisation of n. The second possibility (failure of the inversion step) is again indicated in the variable *invfail* and *test* below. Given $\tilde{P}$, the algorithm calculates the sequence $\{X\}$ where $x_i = V_i(\tilde{P}, 1) \cdot P$. Note that $V_i(\tilde{P}, 1)$ does not need to be calculated explicitly. If $x_{i-1} = V_{i-1}(\tilde{P}, 1) \cdot P$ and $x_i = V_i(\tilde{P}, 1) \cdot P$ then $x_{i+1} = \tilde{P} \cdot x_i - x_{i-1} = V_{i+1}(\tilde{P}, 1) \cdot P$. An algorithm for factorising $n = pq$ can now be sketched as follows:

Input: n, $P = (x, y)$, a, $\tilde{P}$, B; *Output: "success",p,q; or "fail"*

---

set $b = y^2 - x^3 - ax \bmod n$;
set $x_0 = 2 \times P$;
set $x_1 = \tilde{P} \times P$;
set start $= 0$;
repeat
  set $x_{i+1} = \tilde{P} \times x_i - x_{i-1}$; *(* This sets also invfail and test *)*
  if not invfail then set test $= gcd(xcoord(x_{i+1}) - xcoord(x_{start}), n)$;
until test $\neq 1$ *or invfail or* $i > B$;
If test $\neq 1$ *and test* $\neq n$ *then Output("success",test,n:test); else Output("fail");*

Prevention of the attack:
The comments for the prevention of this attack are now similar to those of the previous subsection.

3 Comparison with Pollard's ρ Method

Observe the similarity of the algorithm in Sections 2.2 and 2.3, for $e = 2$ and some instances of Pollard's ρ method, [Pollard75]. If we compare these three methods for factorising an RSA modulus $n = pq$ we have:

$x_0 = seed$, $x_{i+1} = x_i^2 + c \bmod n$, $c \neq 0, -2$,	Pollard's ρ method
$x_0 = seed$, $x_{i+1} = x_i^2 \bmod n$,	algorithm in 2.2
$x_0 = seed$, $x_{i+1} = x_i^2 - 2 \bmod n$,	algorithm in 2.3

Note that Pollard's ρ method requires $c \neq 0, -2$ while the other two algorithms use exactly these values of c. A contradiction? Not according to the authors. The idea behind Pollard's ρ method is to construct a cyclic sequence with some *random* properties. It can be shown that due to these random properties, factorisation of n is obtained after $O(\sqrt{p})$ or $O(\sqrt{q})$ steps (whichever is smaller). The algorithms in Section 2.2 and 2.3 try to exploit some anticipated *structure* of p and/or q in order to achieve factorisation – a different scenario.

4 Conclusion and Future Research

Strong RSA Primes A strong RSA prime until now was a prime p where (1):

- $p-1$ has a large factor, say t;
- $p+1$ has a large factor;
- $t-1$ has a large factor.

Applying our generalised cycling attacks described above, we obtain the following symmetric conditions. A strong RSA prime is a prime p where (2):

- $p-1$ and $p+1$ both have a large factor, say t and w;
- $t-1$ and $t+1$ both have a large factor;
- $w-1$ and $w+1$ both have a large factor.

There is no reason to prefer (1) to (2) (see also below). The attacks that give rise to (2) have the same order of complexity as the attacks that imply (1). However, it is certainly debatable to drop all of these conditions, that is, (1) and (2) (or only insist on $p-1$ and $p+1$ having a large factor). This is because (1) and (2) offer no protection against the elliptic curve method, [Lenstra87] and the EC generalised cycling attacks presented in this paper. Moreover, primes that satisfy (1) and/or (2) might be too "sparse" and/or "not random enough" – a disastrous scenario from an information security point of view.

At this moment the attacks described in this paper are of theoretical value only. We do not anticipate that the attacks pose a practical threat to RSA if the primes are chosen large enough. In future research we will (i) quantify primes of a given size that are susceptible to generalised cycling attacks and therefore throw more light on the above discussion; (ii) examine and discuss the EC generalised cycling attacks and variants thereof.

Maurer's Theorem 6, [Maurer95], Does not Apply Maurer's Theorem 6 in [Maurer95] does not apply to the attacks presented here. [Maurer95], Page 148 states that "Iterated t–fold encryption in an RSA cryptosystem reveals the plaintext x if and only if $x^{e^u} \equiv x(\bmod m)$ for some $u \leq t$, i.e., if and only if $e^u \equiv 1(\bmod ord_m(x))$ for some $u \leq t$" (remark: m in [Maurer95] is the RSA modulus, that is, n in our paper). Page 149 concludes "Theorem 6 illustrates that, in order to prevent decipherability by iterated encryption, the condition, suggested by Rivest [78] and other, that $p' - 1$ (where p' is the largest factor of $p-1$) must also have a very large prime factor p'', is unnecessary." The scenario considered in [Maurer95] corresponds to Case 1 in the abstract of our paper. Case 2 (cycling attacks and EC generalised cycling attacks) is *not* considered. Hence, Theorem 6 covers only Case 1 and does not apply to our attacks.

References

[AlyMue96] H. Aly and W.B. Mueller, Cryptosystems based on Dickson polynomials, *PRAGOCRYPT'96 preproceedings*, 493–503, 1996.

[AndVau96] R. Anderson and S. Vaudenay, Minding your p's and q's, *ASIACRYPT'96*, Springer LNCS 1163, 26–35, 1996.

[BBS86] L. Blum, M. Blum and M. Shub, A simple unpredictable pseudorandom number generator, *SIAM Journal on Computing*, 15, 364–383, 1986.

[BleBosLen95] D. Bleichenbacher, W. Bosma, A.K. Lenstra, Some remarks on Lucas-based cryptosystems, *CRYPTO'95*, Springer LNCS 963, 386–396, 1995.

[CCY95] C.Y. Chen, C.C. Chang, W.P. Yang, A $\lambda(p-1)$ method of factoring RSA's modulus, *Cryptography Policy and Algorithms Conference, CPAC'95 preproceedings, Brisbane 1995*, 225–231, 1995.

[GysSeb98] M. Gysin and J. Seberry, Generalised cycling attacks on RSA, technical report, TR 1998/1, 1998.

[Huber91] K. Huber, Some considerations concerning the selections of RSA moduli, *EUROCRYPT'91*, Springer LNCS 547, 294–301, 1991.

[Koblitz87] N. Koblitz, Elliptic curve cryptosystems, *Mathematics of Computation*, Vol 48, 177, 203–209, 1987.

[KMOV92] K. Koyama, U. Maurer, T. Okamoto and S.A. Vanstone, New public-key schemes based on elliptic curves over the ring Z_n, *CRYPTO'91*, Springer LNCS 576, 252–266, 1992.

[Lenstra87] H.W. Lenstra, Factoring integers with elliptic curves, *Annals of Mathematics* 126, 649–673, 1987.

[Lucas1878] F.E.A. Lucas, Théorie des fonctions numériques simplement périodiques, *American Journal of Mathematics*, 1, 184–240/289–321, 1878.

[Maurer95] U.M. Maurer, Fast generation of prime numbers and secure public-key cryptographic parameters, *Journal of Cryptology*, Vol. 8, 3, 123–155, 1995.

[MeyMue96] B. Meier and V. Mueller, A public-key cryptosystem based on elliptic curves over Z/nZ equivalent to factoring, *EUROCRYPT'96*, Springer LNCS 1070, 49–59, 1996.

[Menezes93] A.J. Menezes, *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers, Massachusetts, USA, 1993.

[MenOorVan97] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, USA, 1997.

[Pollard74] J.M. Pollard, Theorems on factorisations and primality testing, *Proceedings of the Cambridge Philosophical Society*, 76, 521–528, 1974.

[Pollard75] J.M. Pollard, A Monte Carlo method for factorisation, *Nordisk Tidskrift för Informationsbehandling (BIT)*, 15, 331–334, 1975.

[Rabin79] M.O. Rabin, Digitalized signatures and public-key functions as intractable as factorization, MIT/LCS/TR-212, MIT Laboratory for Computer Science, 1979.

[Riesel85] H. Riesel, *Prime Numbers and Computer Methods for Factorization*, Progress in Mathematics, Vol 57, Birkhaeuser, Boston, 1985.

[RivShaAdl78] R. Rivest, A. Shamir and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM*, 21, 2, 120–126, 1978.

[SmiSkin94] P. Smith and C. Skinner, A public-key cryptosystem and a digital signature algorithm based on the Lucas function, *ASIACRYPT'94*, pre-proceedings, 298–306, Wollongong, 1994.

[Stinson95] D.R. Stinson, *Cryptography Theory and Practice*, CRC Press, Boca Raton, USA, 1995.

[Vajda89] S.Vajda, *Fibonacci & Lucas Numbers and the Golden Section: Theory and Applications*, Halsted Press, John Wiley and Sons, New York, 1989.

[Williams82] H.C. Williams, A $p+1$ method of factoring, *Mathematics of Computation*, 39, 225–234, 1982.

A Appendix

A.1 Lucas Sequences

Let $P \geq 3$, $V_0 = 2$, $V_1 = P$ and for $n \geq 2$, $V_n = PV_{n-1} - QV_{n-2}$. This sequence is called a Lucas sequence. (Often instead of only writing V_n we write $V_n(P,Q)$. The whole sequence $\{V\}$ will be denoted by $V(P,Q)$.) The following properties (amongst many others) are elementary and well known:

1. If α and β are distinct roots of the polynomial $x^2 - Px + Q = 0$, then $V_n = \alpha^n + \beta^n$.
2. $V_n(V_k(P,Q),Q^k) = V_{nk}(P,Q)$. In particular, if $Q = 1$, then $V_n(V_k(P,1),1) = V_{nk}(P,1) = V_k(V_n(P,Q),1)$. This property forms the basis for many RSA and ElGamal type cryptosystems.

Note that if $Q = 0$ then $V_n = P^n$ for $n \geq 1$. In other words, Lucas sequences can be looked at as generalised exponentiation. This property and the above mentioned property (2.) is the basis for many RSA and ElGamal type cryptosystems, cryptographically secure pseudo-random bit generators (CSPRBG), and factorisation algorithms based on Lucas sequences.

Let us now try to calculate the period $\pi_{V(P,1),p}$ of $V(P,1) \bmod p$, p prime. Let $D = P^2 - 4Q = P^2 - 4$, $D \neq 0$ and assume D is square-free. From above we have

$$\alpha = \frac{P + \sqrt{D}}{2}, \quad \beta = \frac{P - \sqrt{D}}{2}$$

and using Fermat's theorem in the quadratic field $Z_p[\sqrt{D}]$

$$\alpha^p \equiv (\frac{P + \sqrt{D}}{2})^p \equiv \frac{1}{2^p}(P^p + \sqrt{D}^p) \equiv \frac{1}{2}(P + \sqrt{D}^p) \bmod p.$$

Since

$$\sqrt{D}^p \equiv (D^{\frac{1}{2}})^p \equiv D^{\frac{p-1}{2}} D^{\frac{1}{2}} \equiv (D|p)\sqrt{D} \bmod p,$$

we obtain

$$\alpha^p \equiv \begin{cases} \alpha \bmod p & (D|p) = 1 \\ \beta \bmod p & (D|p) = -1 \end{cases}$$

and similarly for β

$$\beta^p \equiv \begin{cases} \beta \bmod p & (D|p) = 1 \\ \alpha \bmod p & (D|p) = -1 \end{cases}$$

It can be shown that,

$$Z_p[\sqrt{D}] \simeq \begin{cases} GF(p) & (D|p) = 1 \\ GF(p^2) & (D|p) = -1 \end{cases}$$

This property allows us to calculate $V_{p-1}, V_p, V_{p+1}, V_{p+2} \bmod p$ for the two cases $(D|p) = 1$ and $(D|p) = -1$. The values are shown in Table 1. Note that $V_p(P,1) \equiv P \bmod p$. This property can be used for probablistic primality tests

	V_{p-1}	V_p	V_{p+1}	V_{p+2}
$(D\|p) = 1$	2	P	V_2	V_3
$(D\|p) = -1$	V_2	P	2	P

Table 1. Some values of $V(P,1) \bmod p$.

based on Lucas sequences.

Since $V_0 = 2$, $V_1 = P$ and the sequence is fully determined by its last two elements we now have for $(D|p) = 1$, $V(P,1) \bmod p$ repeats itself after at most $p-1$ steps; and for $(D|p) = -1$, $V(P,1) \bmod p$ repeats itself after at most $p+1$ steps. More precisely, $\pi_{V(P,1),p} \mid p - (D|p)$.

$V(P,1) \bmod p$ is symmetric. More precisely, $V_i \equiv V_{\pi_{V(P,1),p}-i} \bmod p$. This can be seen as follows. $V_{i-1} = PV_i - V_{i+1}$. Therefore, $V_{\pi_{V(P,1),p}-1} \equiv PV_{\pi_{V(P,1),p}} - V_{\pi_{V(P,1),p}+1} \equiv PV_0 - V_1 \equiv P \equiv V_1 \bmod p$ as claimed. The proof that $V_i \equiv V_{\pi_{V(P,1),p}-i} \bmod p$ follows now by induction on i.

The calculation of the period $\pi_{V(P,Q),p}$ of $V(P,Q)$, $Q \neq 1$ can be done similarly, and it can be shown that, in this case, $\pi_{V(P,Q),p} \mid p^2 - 1$.

It is important to realise that $V_k(P,1)$ (and in general $V_k(P,Q)$) can be calculated in $O(\log k)$ steps by square–and multiply techniques. In other words, one does *not* have to calculate $V_0, \ldots, V_{k-2}, V_{k-1}$ in order to be able to calculate V_k.

One possibility to calculate $V_k(P,1)$ in $O(\log k)$ steps is the following. Consider the 2×2 matrix

$$M = \begin{bmatrix} 0 & 1 \\ -1 & P \end{bmatrix},$$

and the matrix multiplication

$$\begin{bmatrix} a \\ b \end{bmatrix} = M^k \begin{bmatrix} 2 \\ P \end{bmatrix}.$$

It can be shown, [Vajda89], that $a = V_k(P,1)$ and $b = V_{k+1}(P,1)$. M^k (and therefore V_k) can be calculated in $O(\log k)$ steps by square–and multiply techniques.

A.2 Elliptic Curves

We only give a brief introduction into elliptic curves. The reader is referred to [Menezes93] for more details. We only consider elliptic curves over the field F_p or the ring Z_n, where $n = pq$, p, q two primes > 3.

An elliptic curve E is the set of solutions (x, y) to the affine Weierstrass equation

$$y^2 = x^3 + ax + b, \tag{1}$$

together with a point at infinity denoted by $\mathcal{O}$. If E is over F_p or if E is over Z_n, we denote the solutions to (1) by $E(F_p)$ or $E(Z_n)$, respectively. The number of solutions to (1) including $\mathcal{O}$ will be denoted by $\#E(F_p)$ or $\#E(Z_n)$, respectively.

If $4a^3 + 27b^2 \not\equiv 0 \bmod p$, then it can be shown that $E(F_p)$ is an abelian group by defining a suitable operation '+' on its points. $\mathcal{O}$ is the identity element. That is, $P + \mathcal{O} = \mathcal{O} + P = P$. For $P = (x_1, y_1)$, $Q = (x_2, y_2)$, $P \neq \mathcal{O} \neq Q$, $P + Q$ is defined as follows. If $x_1 = x_2$ and $y_2 = -y_1$, $P + Q = \mathcal{O}$. Otherwise $P + Q = R = (x_3, y_3)$, where

$$x_3 = \lambda^2 - x_1 - x_2, \quad y_3 = \lambda(x_1 - x_3) - y_1,$$

and

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & P \neq Q \\ \frac{3x_1^2 + a}{2y_1} & P = Q \end{cases}$$

Let E be an elliptic curve over F_p. It is well known that $E(F_p) \simeq Z_{n_1} \times Z_{n_2}$ where $n_2 \mid n_1$ and $n_2 \mid p - 1$. Furthermore, $\#E(F_p) = p + 1 + t$, where $t^2 \leq 4p$.

We can generalise these addition laws to the case $E(Z_n)$. Clearly, $E(Z_n)$ will not be a group, since the inversion step will not be possible if the denominator and n are not co–prime. Therefore, we call this operation *partial addition*. Whenever partial addition on $E(Z_n)$ is defined, we have for $P = (x, y) \in E(Z_n)$, $P_p = (x \bmod p, y \bmod p) \in E(F_p)$ and $P_q = (x \bmod q, y \bmod q) \in E(F_q)$. Therefore, this partial addition will have the following properties:

- if it is defined, it will yield a new point on $E(Z_n)$;
- if it is not defined, it will lead to the factorisation of n.