The New Master Lock® Speed Dial™ / ONE™ Combination Padlock – An Inside View.

Introduction

Hi, my name is Michael Huebler, I'm a lock collector from Germany, and in March 2008 I was very lucky to receive a sample of a very interesting new padlock that was not yet available on the market – a combination lock from Master Lock® with a unique new user interface and a very clever mechanism inside.

Inspired by other papers on lock designs, such as the Abloy series by Han Fey, I decided to share the details of this nice piece of mechanical engineering with you, and I hope you enjoy reading about them as much as I enjoyed disassembling and analyzing the actual lock.

As of July 2009, the lock has reached the market; it is named "Speed Dial™" in North America and "ONE™" in Europe. Both versions come in four different color combinations, and the North American version has a different knob design.

I would be very happy to receive your comments, corrections, or just a short note whether you found this paper interesting or helpful. You can contact me at mh@TheOpenSourceLock.org.
You can find the most recent version of this paper as well as **a free software program that visualizes the mechanism of this lock for you** at http://www.toool.nl/. Thanks for your interest!

The outside

The lock is about 94mm/3.7" tall (69mm/2.7" without the shackle), 55mm/2.2" wide and 33mm/1.3" thick (28mm/1.1" without the knob); the shackle is about 33mm/1.3" wide, its diameter is about 7mm/0.3" and it is apparently made from hardened steel. The lock body is made from cast metal and painted; as you can see, this prototype comes in a pastel green with white accents. The front and the back of the lock body are held together by 3 screws which will be discussed later.

How to use the lock

This is a combination lock with a new and unique user interface: Instead of turning a wheel to certain numbers, a knob has to be moved on two axes – Up, Down, Left or Right – in a sequence, quite similar to a game console joystick.

The combination sequence can be reprogrammed by the user, and it can have any length (!).

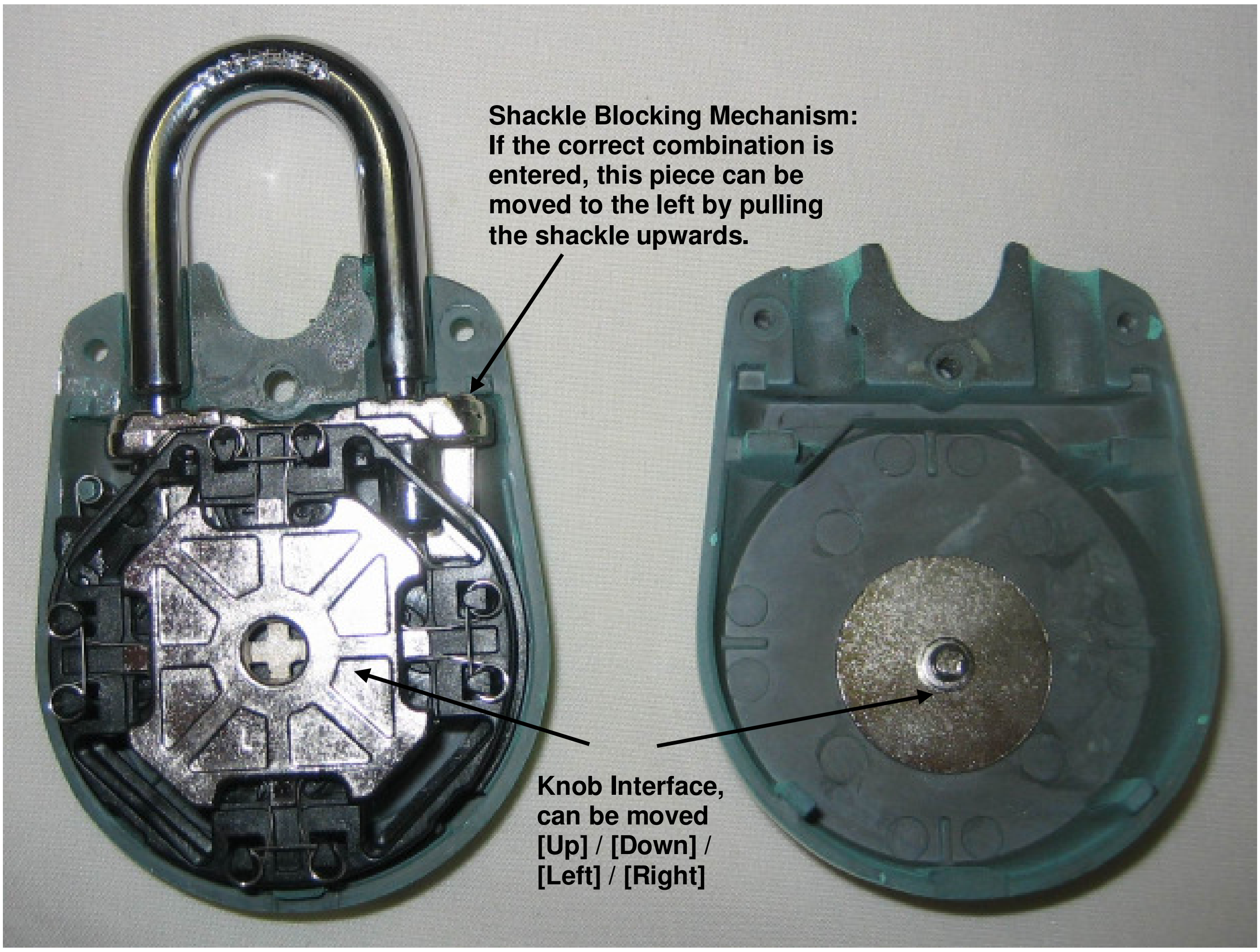
To start, the user pushes the shackle into the lock – this clears the combination inside the lock – and then the user has to move the knob in the correct sequence. The factory combination of the initial samples was [Up, Down, Left, Right].
Once the combination has been entered correctly, the shackle can be pulled out. When the user closes the lock, the shackle has to be pushed in all the way to clear the combination again.

The combination can be reprogrammed by the user quite easily: After the correct combination has been entered, a small pin on the back of the lock can be moved upwards to the “R” position. Then the old combination should be erased by pushing the shackle into the lock, and now a new combination sequence can be entered. Afterwards the small pin has to be moved back, and the lock is now reprogrammed to the new combination.
This works fine, however, there is no way to check if the sequence was entered correctly, other than closing the lock and trying it out – so the reprogramming should be carried out with care in order to avoid a lock with an unknown combination.

The inside – How it works

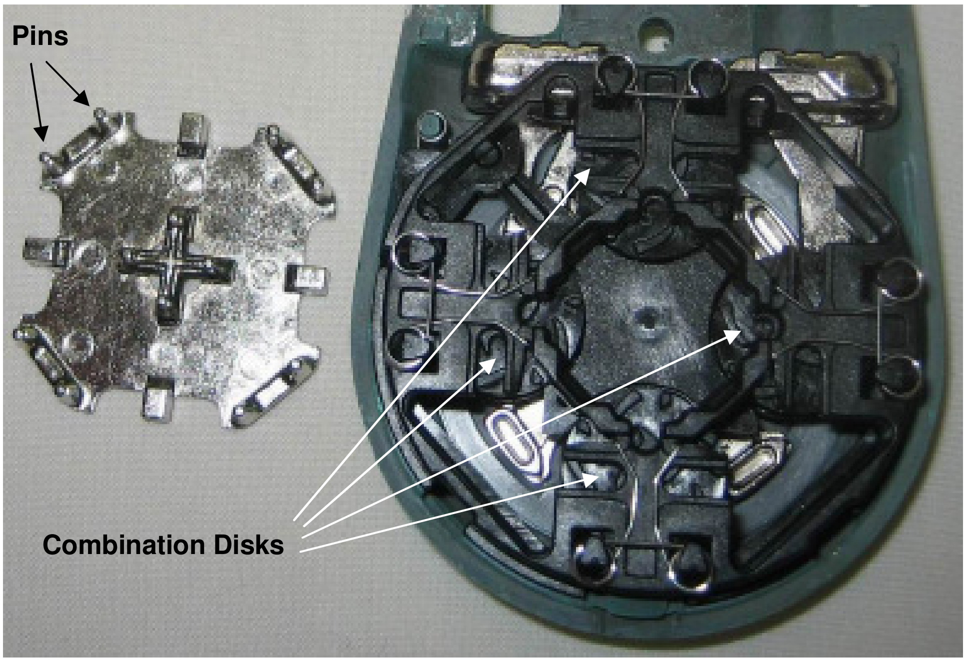
When I received this lock, I immediately knew that I needed to know what’s inside…

After removing the screws, the front of the lock can be removed. The lock is secured by two small cross-head screws and one larger one-way-screw – that’s a screw with a head that accepts torque only clockwise; when turning counter-clockwise in an attempt to remove the screw, the screwdriver will only scratch the head of the screw but not turn it. I had to remove it semi-destructively. All screws are also secured by some type of thread lock glue.

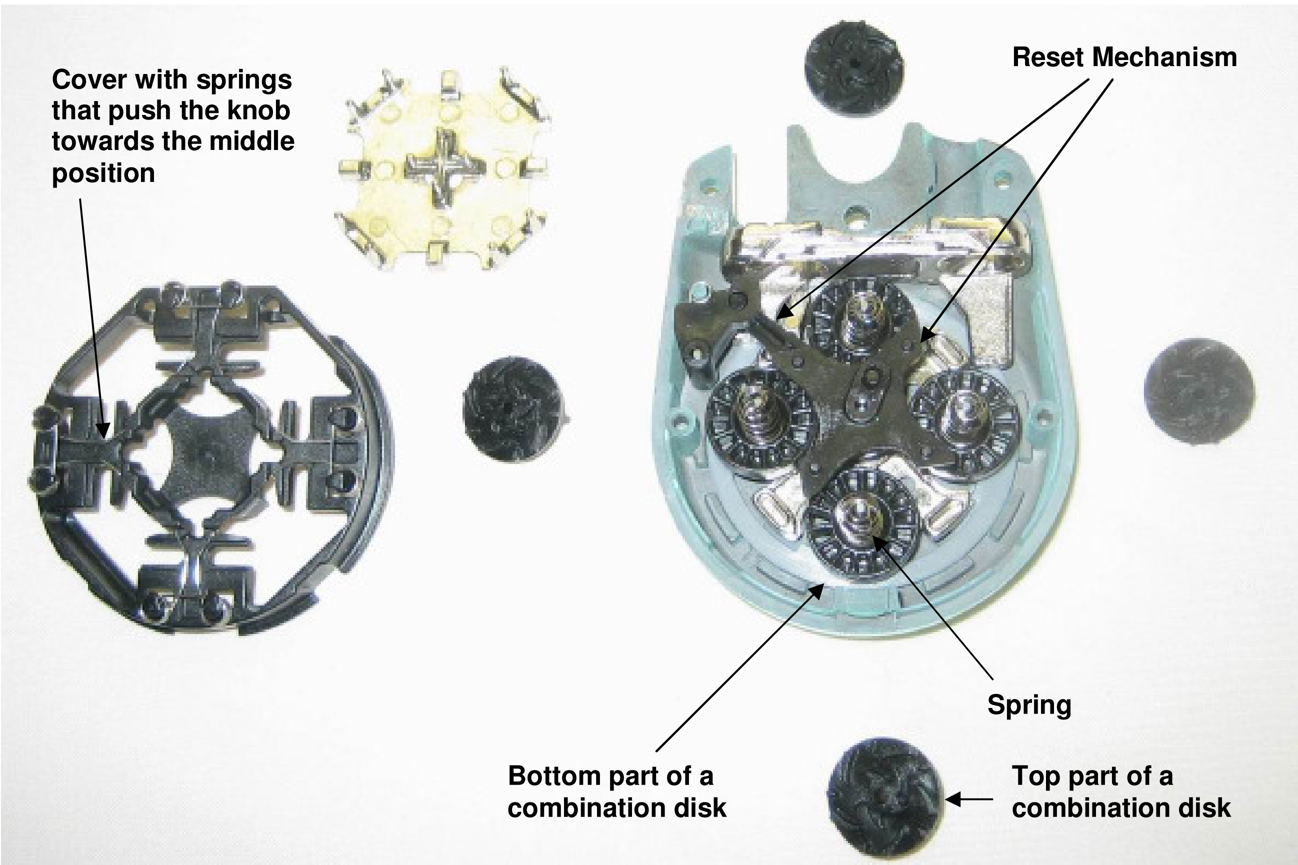


On the left side of the picture, you can see the combination mechanism (black plastic), as well as the knob interface plate (cast metal). This plate is moved when the user moves the knob on the front of the lock.

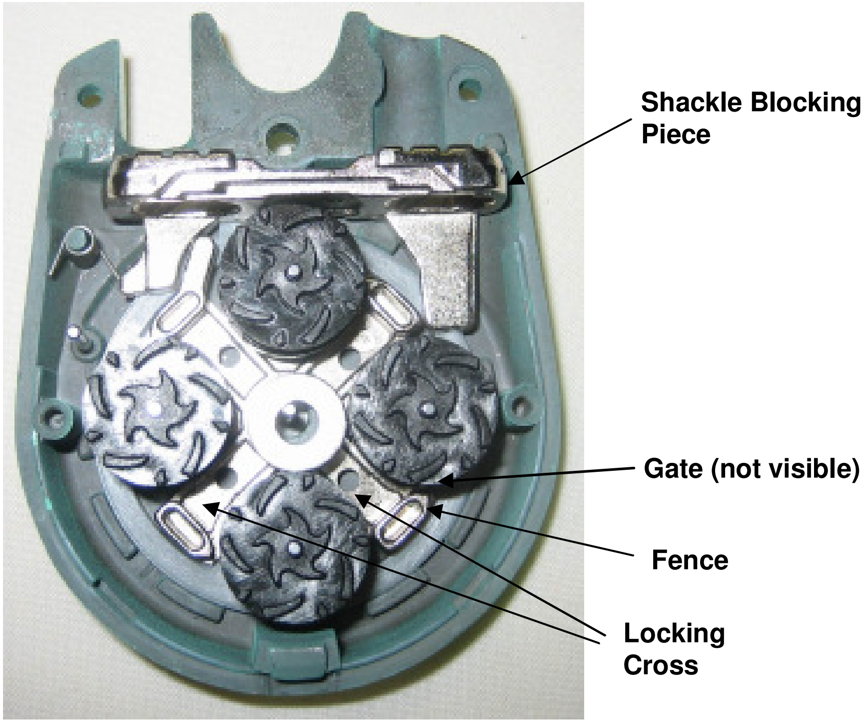
In the next picture, you can see the back of the knob interface plate: It has 12 pins, 4 in the middle of the plate and 2 at each of the 4 corners. On the right hand side you can also see four black combination disks (partially covered) – the pins interface with these disks by means of a lot of cleverly designed control curves. These will be discussed in detail later.



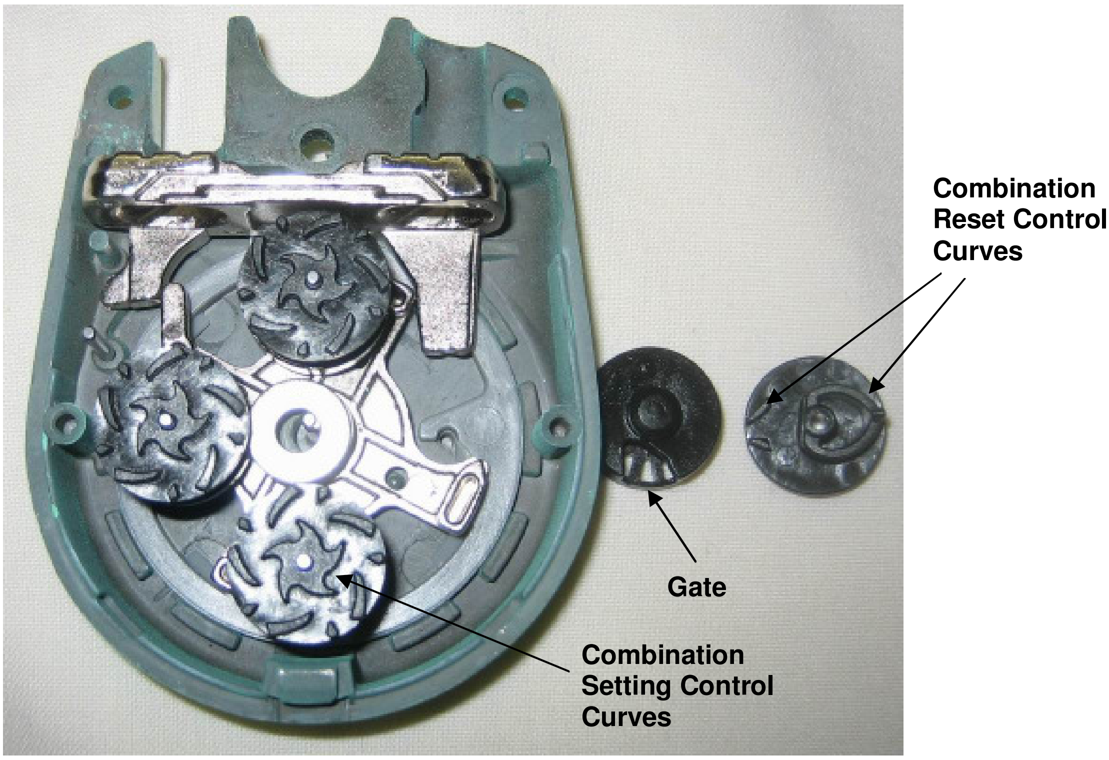
For the picture below, I disassembled the lock even further: On the right you can see the lock body with the bottom parts of the 4 combination disks still in place; as you can see, the combination disks consists of two plastic parts (top and bottom) with a spring in between – they are separated when the user actuates the reprogramming mechanism, then later pressed back together in one of 15 possible relative positions.
You can also see part of the black plastic reset mechanism that resets the combination when the left side of the shackle is pushed into the lock.



Now let's have a look at the mechanism that checks whether or not the correct combination has been entered: When you pull on the shackle, it will try to move the blocking piece to the left by means of angled surfaces. This blocking piece is connected to a cross-shaped gate testing piece – the locking cross – (similar to the 'lever' and 'fence' in other types of combination locks) that will try to turn counter-clockwise into the gates in the bottom parts of the four combination disks.

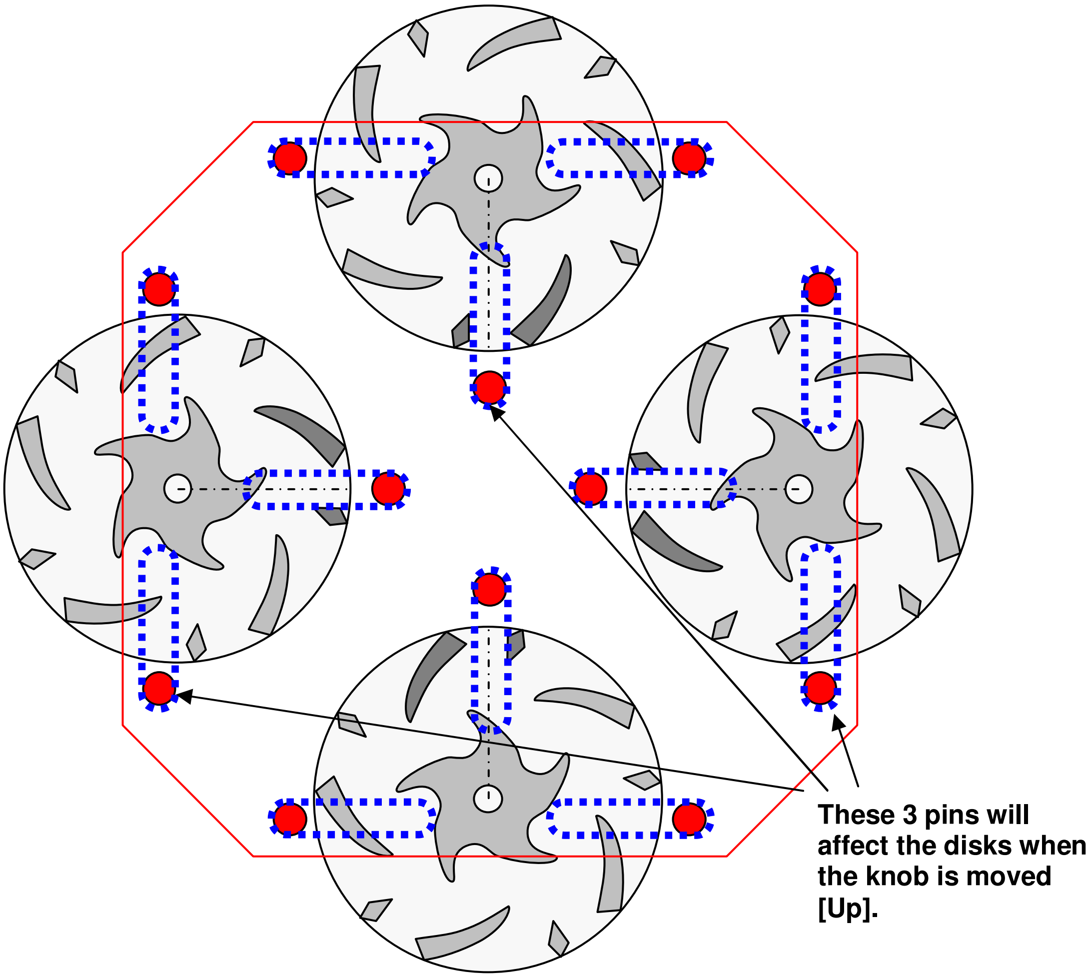


In the next picture, the disks are set to their correct angles φ_x and the locking cross has successfully moved into the gates. On the right you can also see the back of the combination disk parts (bottom and top parts), and the shape of the gate in the bottom part. Finally, take a note of the combination reset control curves on the back of the top part.



But for me the most interesting part is how the sequence of knob movements translates into the setting of the combination disks. How does the lock check the sequence in which the knob is moved? Why does a lock programmed to the sequence [Up, Down] not open with [Down, Up] as well?

To understand this, have a look at the combination disks in their reset position:

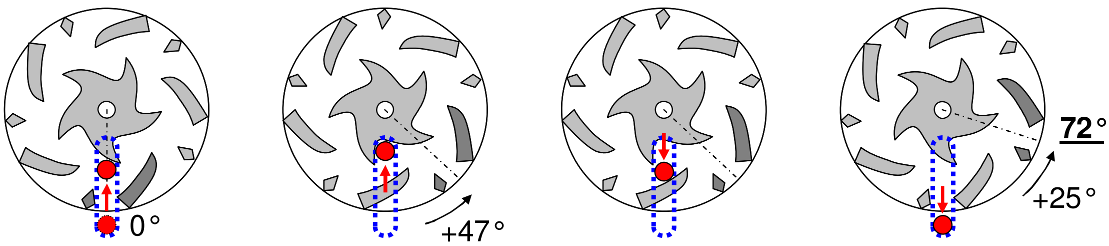


The grey areas show the control curves that interact with the pins of the knob interface plate (shown in red color). You can imagine that moving the knob [Up] will somehow affect the left, the upper, and the right disk, because three pins will push against the control curves of these disks. The lower disk will not be affected when the knob is moved [Up].

But how exactly will the pins affect the disk angles?

For simplification, we will now concentrate only on one disk (the upper disk), but keep in mind that each knob movement always affects three disks.

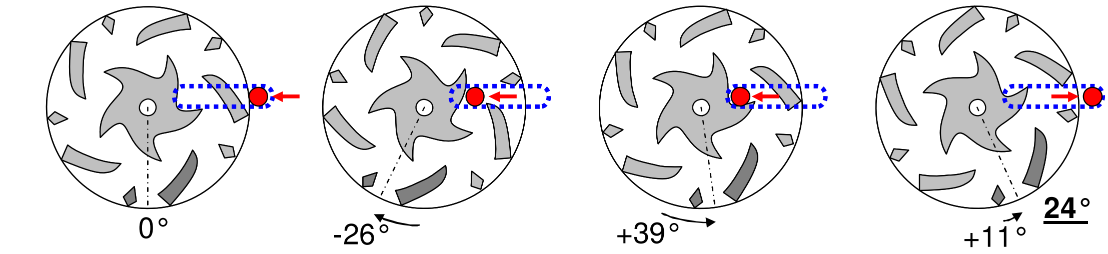
The next picture shows the upper disk while the knob is moved [Up] and then released again:



In this example, we start at the reset position (0°). Then the pin is moved upwards. After it touches the inner control curve, pushing it further upwards will turn the disk counter-clockwise by 47°. Then the knob is released and the pin pushes against one of the outer control curves, turning the disk by another 25° to the final position at 72°.
We note that moving the knob [Up] while the upper disk was initially positioned at 0° will turn the upper disk by 72°.

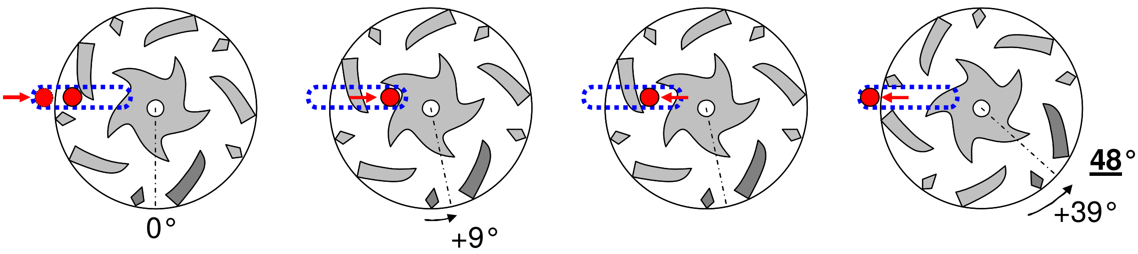
We also note that the end position looks quite similar to the starting position – that's because the disks are symmetrical; the same pattern of control curves repeats 5 times every 72°.

Now let's have a look what happens if the first move in the sequence is not [Up], but [Left]. In that case, another pin moves from the right to the left and then back:



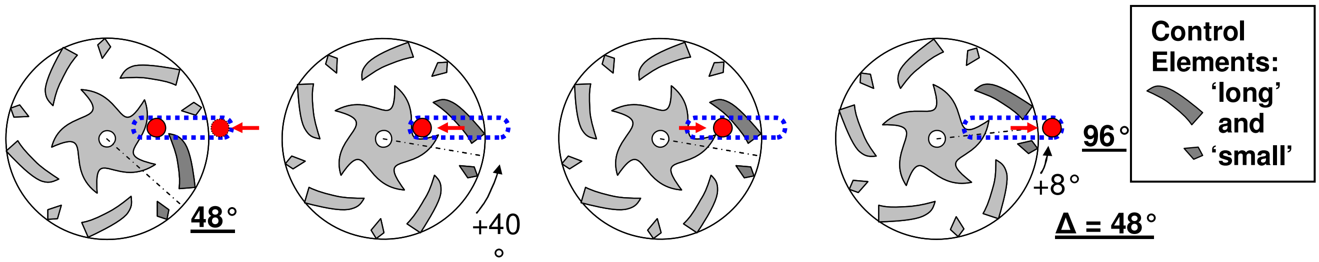
In this case, the disk is first turned clockwise, then counter-clockwise, and the resulting total angle is only 24°.

The 3rd option for the upper disk is a [Right] move:



Here, the disk is turned in two steps to the final position at 48°.

Another example where the disk is *initially not positioned at its reset angle* shows again a different result:



Generally, moving a selected pin in and out will always move the disk by either 24°, 48°, or 72° to the next valid angle in a set of 5 possible 72° sections. This is due to the shape of the control curves: The pin will always leave the disk between a 'long' control element to its left and a 'small' control element to its right.

The set of possible angles depends on the pin being used on the disk: If the bottom pin moves upwards, it will turn the upper disk to an angle of n · 72°, the left pin will turn it to n · 72° - 24° and the right pin will turn it to n · 72° + 24°. The angles will always be multiples of 24° (360° / 15).

To better reflect the rotational symmetries of the disks, we now introduce a ***position index (N; M)***, with the actual angle being calculated as N · 72° + M · 24°. M can have one of three values: -1, 0, or +1, depending on which of the three pins was used; and N corresponds to the 5 possible 72 degree sections mentioned before, so N can have values from 0 to 4.

After the disks have been reset, the index is (0; 0) = 0° for all of them.

This table shows the effect of the knob movements on the upper combination disk for all possible starting positions:

Start			Knob Movement [Right]*			[Up]*			[Left]*		
N	M	Angle	New N	New M	New Angle	New N	New M	New Angle	New N	New M	New Angle
0	-1	-24°	1	-1	48°	0	0	0°	0	+1	24°
0	0	0°	1	-1	48°	1	0	72°	0	+1	24°
0	+1	24°	1	-1	48°	1	0	72°	1	+1	96°
1	-1	48°	2	-1	120°	1	0	72°	1	+1	96°
1	0	72°	2	-1	120°	2	0	144°	1	+1	96°
1	+1	96°	2	-1	120°	2	0	144°	2	+1	168°
2	-1	120°	3	-1	192°	2	0	144°	2	+1	168°
2	0	144°	3	-1	192°	3	0	216°	2	+1	168°
2	+1	168°	3	-1	192°	3	0	216°	3	+1	240°
3	-1	192°	4	-1	264°	3	0	216°	3	+1	240°
3	0	216°	4	-1	264°	4	0	288°	3	+1	240°
3	+1	240°	4	-1	264°	4	0	288°	4	+1	312°
4	-1	264°	0	-1	336°	4	0	288°	4	+1	312°
4	0	288°	0	-1	336°	0	0	0°	4	+1	312°
4	+1	312°	0	-1	336°	0	0	0°	0	+1	24°

* This example refers to the upper disk.

We now understand why the lock can check the sequence in which the combination has been entered:
Moving the knob in one direction will affect 3 disks differently, and how they will be affected *depends on how they have been turned before*.

Number of different combinations

As mentioned before, the lock can actually be programmed with a sequence of unlimited length. It works like a hash function that translates the sequence of knob movements into a final state of the four combination disks; and this state is subsequently checked when the user pulls on the shackle.

While the length of the sequence is unlimited, the 4 combination disks can obviously only have a limited number of different states. To calculate that number, which is also the maximum number of different combinations, we look at the states the disks can have.
At first glance, one could assume that there are 15 ^ 4 = 50,625 possible states (because of the 4 disks with 15 positions each); however, there are in fact certain constraints due to the arrangement of the disks and the knob interface: The last knob movement restricts the possible values of the 3 disks that have been affected.

We look at the case in which the last knob movement was [Up] – this will leave the left combination disk in a state of (x; -1), the upper disk in a state of (y; 0), and the right disk in a state of (z; +1); x, y, z can all have 5 different values (0 to 4). The lower disk can have any of the 15 possible states. This allows for (5 ^ 3) · 15 = 1,875 states.
There are 4 possible last knob movements, and also including the reset state, we calculate 4 · 1,875 + 1 = 7,501 valid states, and thus 7,501 possible combinations.

Further investigation[1] shows that with a maximum of 11 operations (not counting the first reset operation), all states can be reached. Longer sequences can be used, but will always result in states that can also be reached by shorter sequences.
Within the space of sequences with a maximum of 11 operations, only 5 sequences lead to unique states; these are the reset-only sequence (this is trivial and caused by the fact that no operation will lead back to all disks having the (0; 0) position); and [Up, Right, Right, Left, Left, Down, Up], as well as the 3 other sequences that can be derived by turning everything by 90°, 180° and 270°.

Examples of sequences that lead to the same result are [Down, Right, Down] and [Left, Right, Down][2]; or [Left, Left, Right], which is one of the replacements for the sample combination [Up, Down, Left, Right].

[1] The author has used a spreadsheet and a simulation program that could be made available upon request.
[2] This is an interesting example of two related sequences: Even with two different first knob movements, after the third movement the same end result will be reached.

Some more statistics are shown in the table below:

Number of Operations	States that have not been reached by shorter sequences	States that can be reached by no other sequence*	States that can be reached by only 1 other sequence*
0	1	1	0
1	4	0	0
2	16	0	0
3	60	0	4
4	168	0	0
5	396	0	0
6	816	0	8
7	1,448	4	8
8	1,984	0	4
9	1,796	0	8
10	708	0	8
11	104	0	12
Sum	7,501	5	52

* only sequences with up to 11 operations have been taken into account

Manipulation techniques

These are my thoughts on using non-destructive manipulation techniques on this lock:

Shimming the shackle is not possible due to the shackle locking mechanism.
Using extremely thin wires (e.g. those used in some of John Falle's decoders) to probe the gate positions might be possible, but is difficult and requires complicated tools. Such an attack therefore seems to be impractical for the obviously intended uses cases, which are not top secret military installations, but school lockers etc.
The glued-in one-way-screw effectively prevents non-destructive opening of the lock, however, after the screw has been drilled out, it can of course be replaced by a new one.

If one of the disks has been turned to the correct gate position, the shackle can be pulled out slightly more than before. The user can also feel this effect, because the knob cannot be moved all the way into a direction that would affect that disk. Using a precision measurement instrument and software that visualizes the internal state of the lock, it can be decoded in a process that needs practice and takes quite some time.

Overall I believe that this lock provides quite adequate protection for the mentioned use cases.

Acknowledgements

I would like to thank Julian Hardt and Peter Field, as well as Yehonatan Knoll (the inventor of this lock) for their helpful comments, and of course the Master Lock® Company for providing me with samples of this very interesting new product.

Disclaimer

The opinions expressed here are those of the author only; the author is not affiliated with the Master Lock® Company in any way; the Master Lock® Company or the author's employers have nothing to do with this document. All trademarks are the property of their owners. Some of the concepts and techniques mentioned in here are protected by intellectual property rights such as patents. The information was derived only from the analysis of two locks and might be incomplete and / or contain errors. The author gives no warranty and accepts no liability whatsoever concerning this document.