**Lockbeepers present:**

# An educating evening with the Burgwächter TSE 3000 Basic Code

## Summary

The Burgwächter TSE 3000 Basic Code is an entry-level electronic door lock with code entry that is sold throughout Europe. We got ours in mint condition for about 80 Euro on ebay, normally it is about twice as much. The lock provided a long evening of good old reverse engineering fun for about four people, so its certainly good value for the money.

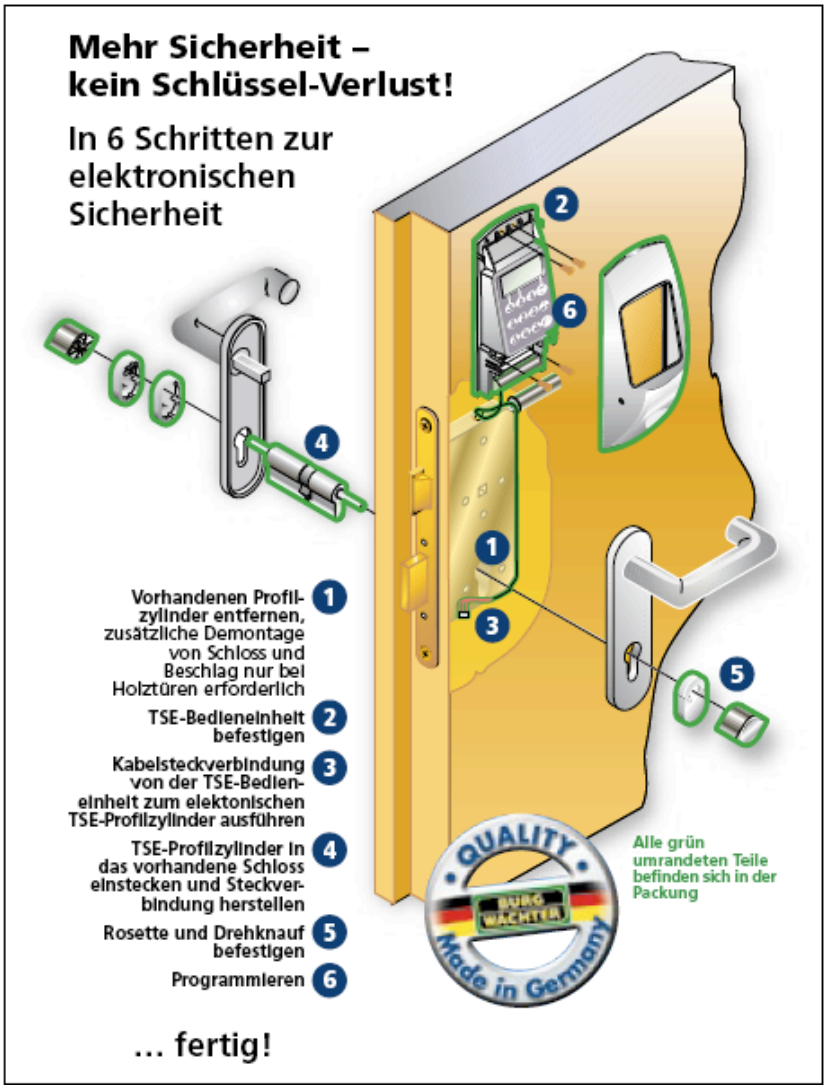
We found two simple, real-world ways to defeat the lock which can be applied covertly if desired. While Burgwächter did a reasonably good job in some aspects of the lock design, they screwed up in others. Some of the protocols, electronics and components are probably used in the other Burgwächter TSE lock products, so this paper may be a good starting point if you plan to toy around with the other Burgwächter TSE lock variants.

Our overall security rating is: **nice educational toy**

**About Lockbeepers:**
Lockbeepers is a loose group of recreational hackers vaguely associated with the Chaos Computer Club, Germany. We aim to advance the art of lockpicking into the 21. century by applying basic reverse engineering and computer security techniques to widely used electronic locks. Since the lock industry generally takes a rather unrelaxed attitude towards people pointing out flaws in their products for fun, we prefer to stay anonymous for the moment.

# Overview

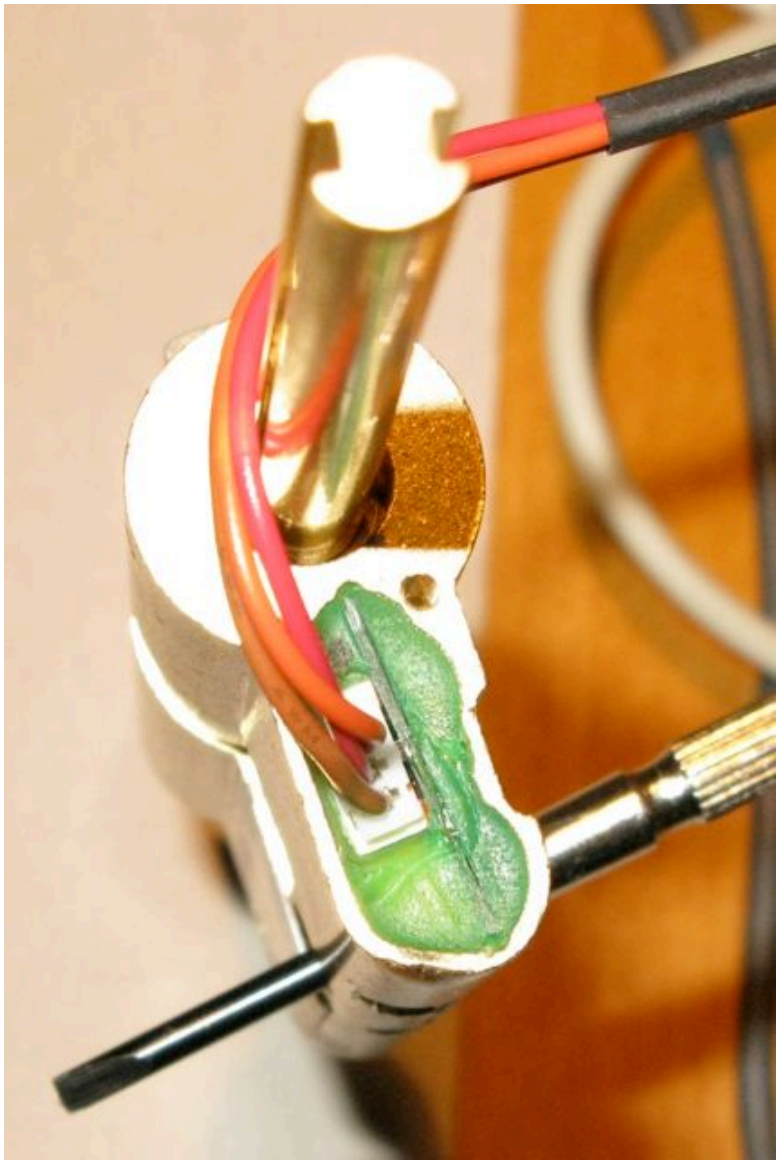
The TSE 3000 BASIC consists of two main parts, the cylinder that replaces your standard mechanical euro-cylinder and the keypad for entering the code. The single opening code is limited to 6 numeric digits (There is a Premium version named TSE 3003 that works with up to 8 digits and multiple users). The keypad also houses the batteries to power the system, so they can be changed from the outside. The keypad is connected to the cylinder by a three-wire cable. (There are also TSE 3000 variants with a wireless connection between the lock and the keypad, that claim to have AES encryption for the radio link, but we haven´t had our hands on one of those yet.)




*Overview on the lock. Source: Burgwächter Website.*

Integrated into the cylinder is a small PCB that receives the numbers typed into the keypad by cable. If the correct number is entered, a motor in the rear part of the cylinder is powered. The motor drives a little lever that engages the axle connected to the knob on the outside of the door, so the knob can be turned by the user to unlock the door. The knob on the inside of the door is always engaged, so it can always be used to open or lock the door from the inside without entering the code. The cylinder does not turn by itself, so in order to lock the door from the outside, you need to turn the knob too.

# First look at the cylinder



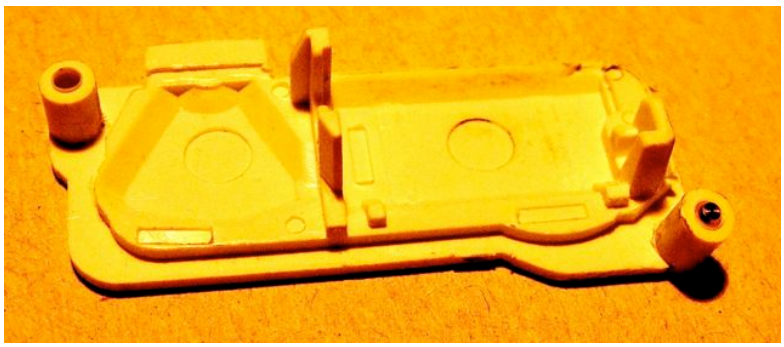
On the front side of cylinder, the connector to the keypad is visible was well as the soft green ectoplasma goo that is used to glue the small PCB into the cylinder.

The first thing that struck us as odd is that the PCB in the lock (which later turns out to hold all the magic) is facing toward the outside of the door. When mounted, it is covered by a metal-color sprayed plastic cover with a hole for the knob axle. The practical reason is probably that it is much easier to route the cable to the keypad this way, without the need to drill extra holes through the door.

On the rear side of the cylinder is the housing for the motor. It sits under a white plastic cover. The cable between PCB and motor is running in a groove on the underside of the lock.

Left is inside, right is outside of door. Visible is the green ectoplasma goo in the hole to the PCB and some wool-like stuff they used to plug the cable hole at the motor cavity.

The mechanics that are driven by the motor to engage the outside axle are made from some durable plastics, so no magnet-of-death-attack here. When the motor turns, the spiral engages the slider on top and moves it to the left in a curious angular motion.



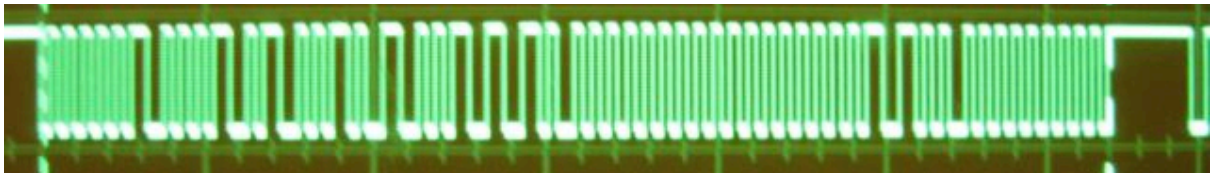
The plastic cover over the motor housing is cleverly engineered to save mechanical parts. Clearly visible is the angular motion cavity.

## Some measurements

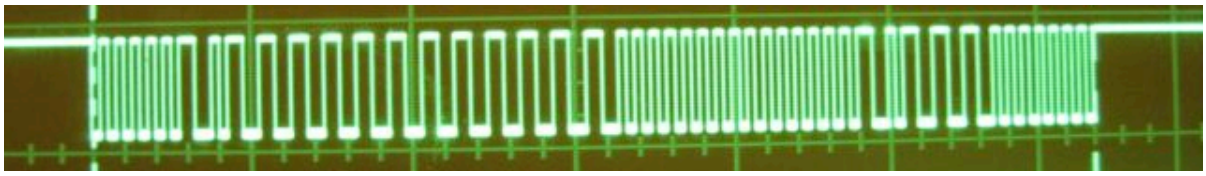
After carefully reassembling the cylinder we needed to figure out the communication between keypad and cylinder. The three wire cable yielded its secrets rather quickly.

| Color | Usage |
|---|---|
| Orange | GND |
| Red | Vcc |
| Brown | Data |

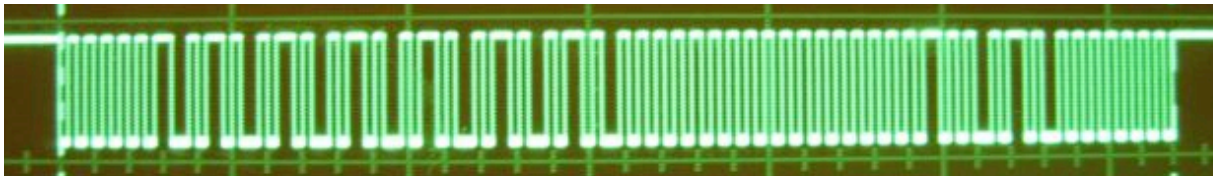
The Data on the brown cable turned out to be a moderately interesting bidirectional 1-wire protokoll that is default high against GND. The datagrams are sent as 64 ms long bursts, with a short high-low change taking place in 1 ms. The voltage on the wire is approximately TTL-level.
Upon first key press, a “wake up” datagram is sent from the keypad to the cylinder. It is always the same and is acknowledged from the cylinder with a similar datagram, that is also always the same. After all six digits have been entered on the keypad, a datagram with the numbers is sent to from the keypad to the cylinder, containing a preamble, the six digits and a checksum.



Datagram for code 123456



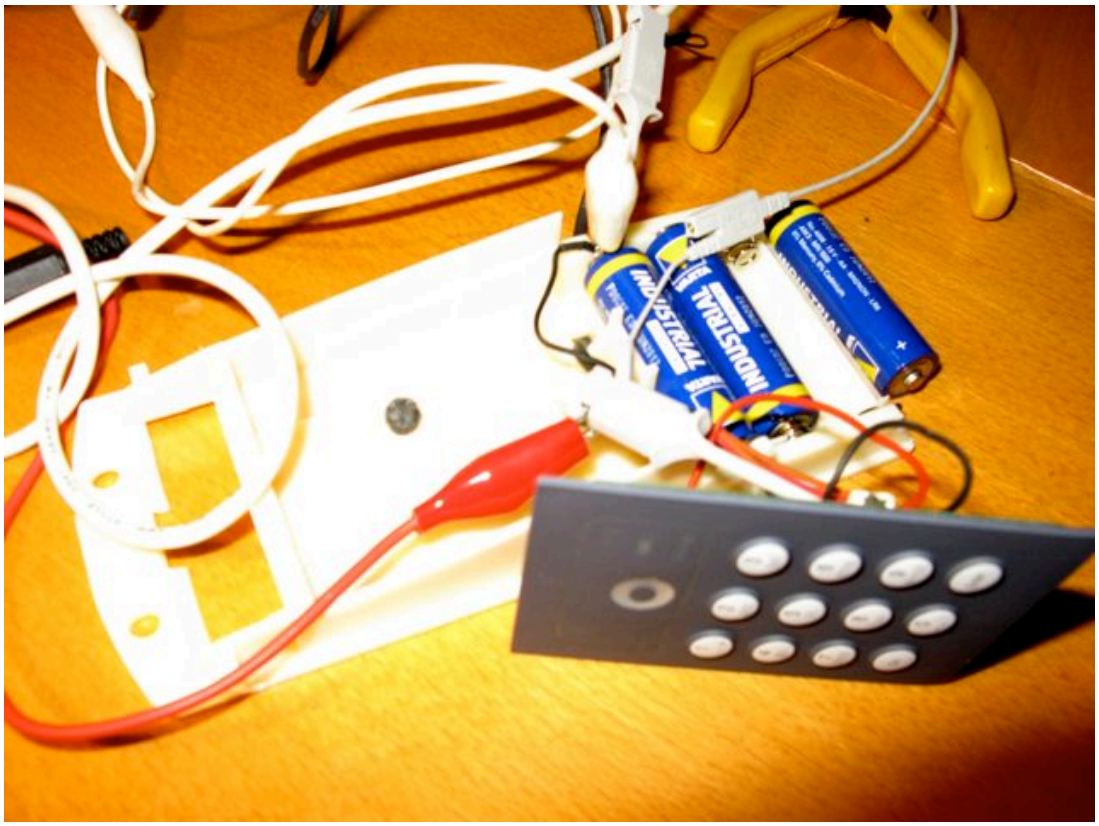
Datagram for code 555555



Datagram for code 666666

The encoding used on the wire is straightforward Manchester with high-low transition representing a zero. In order to get the data easier into the computer for decoding (writing down bits from oscilloscope pictures is not an option here), a line-in audio cable was adapted with a 220nF capacitor on the tip, and connected to the brown wire. (Beeeep!) The signal

turned out just fine and sounded funky, so a little decoding script was employed to get to the bits.




*Measurement setup. The signal is captured at the keypad for convenience reasons.*

**A typical dialog looks like this:**

Keypad to cylinder: *Hey! Wake up! Someone pressed a key!*

```
2.29631s: 0000000100000000000000000000000000000000000000000010000000000000
```

Cylinder to keypad: *Hmmbrmbl... I am awake now.*

```
2.42731s: 0000000100000000000000000000000000001011000000000011011000000000
```

Keypad to cylinder: *Hey! The dude keyed in 123456!*

```
5.71056s: 0000001000010010001101000101011000000000000000000010111000000000
```

Cylinder to keypad: *Looks about right... I am open now.*

```
5.84069s: 0000001000000100000000000000000000000000000000000010000000000000
```

For better overview, here the same dialog again in hexadecimal.

```
2.29631s: 01 00 00 00 00 00 20 00                                ...... .
2.42731s: 01 00 00 00 0b 00 36 00                                ......6.
5.71056s: 02 12 34 56 00 00 5c 00                                ..4V..\.
5.84069s: 02 04 00 00 00 00 80 00                                ........
```

The lock is not entirely stupid. There is a timeout after three false tries, which are counted in the cylinder. The timeout logic also resides in the cylinder. When the lock is in timeout mode, the keypad sends a status request every second to the cylinder, who then signals to the

keypad when the timeout is over, which then switches off the “blocked”-LED. Interestingly on reprogramming, where you need to enter the new code twice, the check that verifies that you really entered the same code twice is performed in the keypad, as there is no communication at this time.

**So we conclude:**

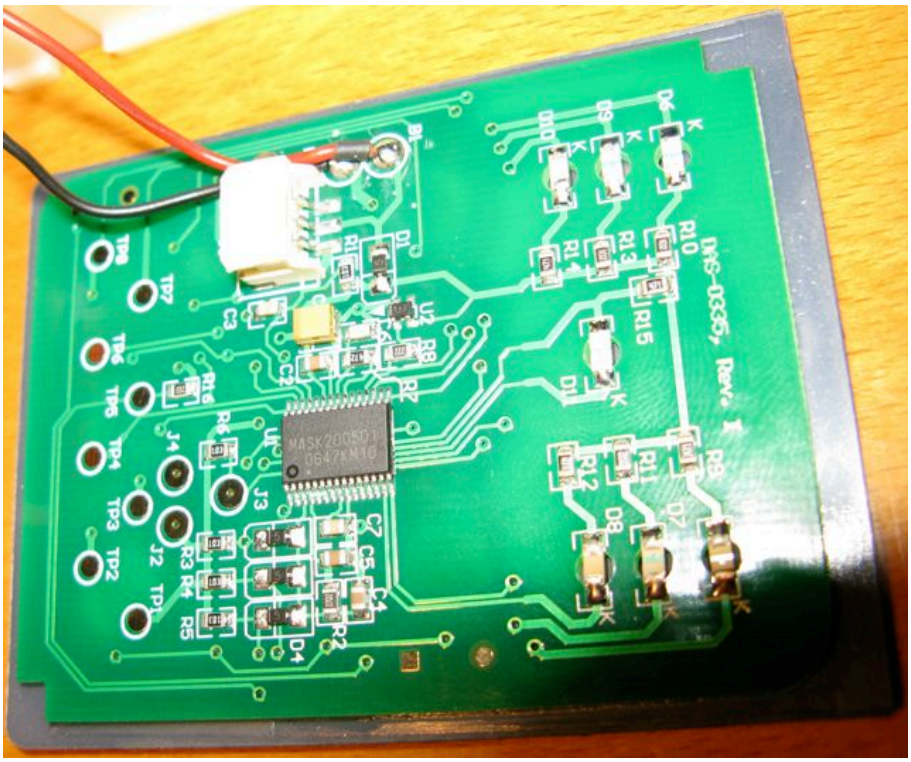
1. There is no encryption or obfuscation on the wire between keypad and cylinder
2. The protocol is simple, straightforward and easy to decode
3. There are no measures against replay attacks (the datagrams are always the same for the same code)

A realistic attack is to place a microcontroller into the keypad enclosure that listens on the brown wire for datagrams. The microcontroller stores the most current code-entry datagram from the keypad that is followed by a “opened successfully” datagram from the cylinder (so we store only the good stuff). When a special preprogrammed keycode is detected (which is the malicious user requesting entry), the stored code-entry datagram is replayed after a certain delay to avoid lock confusion. The program is straightforward and simple. The microcontroller can easily live off the power provided by the keypad-batteries.

As we wanted to look into a few other aspects of the TSE 3000 (and the resident mathemagican got offensively bored for lack of crypto to analyze at this point), we did not dive further into the protocol details and also did not bother to actually implement the replay attack. This is left as an exercise to the reader for a dark, cold winter evening.

# Disassembly and PCBs

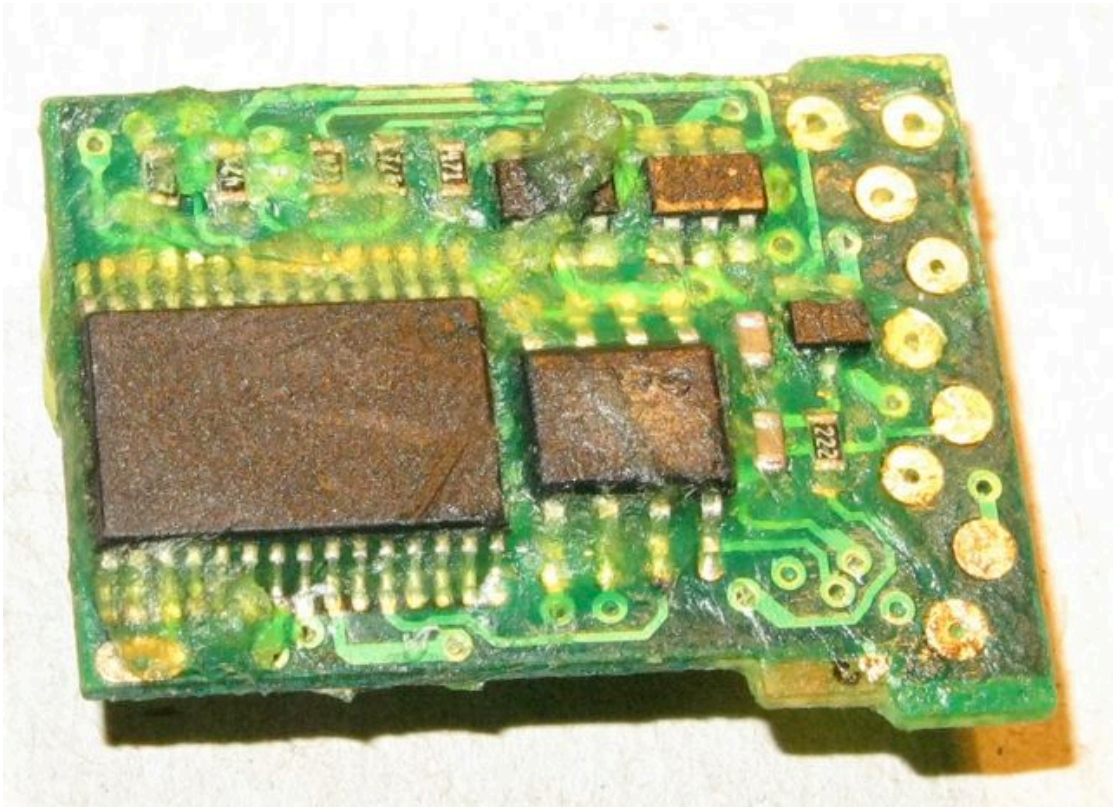
The PCB of the keypad is easy to access. Just open the keypad cover as directed in the manual for battery change. The keypad PCB is rather unspectacular. The functionality resides in a 30-pin chip labeled MASK200501, which is probably a PROM-based custom version of a popular microcontroller. The red and black wires go to the batteries. The rest of the PCB is LEDs, keypad contacts, connector for the cable to the cylinder and some glue electronics.



*Keypad PCB*

To get to the small PCB in the cylinder was more interesting. The green ectoplasma goo is surprisingly soft and has the consistency of tough jelly or soft hot glue (which we suspect is what Burgwächter uses). The method recommended by Buddha is to carefully scrape it away with a small screwdriver, pliers or whatever is at hand. Remember to start at the bottom of the cylinder, and unplug the cable to the motor first! After about 5  minutes of scraping you will have removed all of the goo that you can reasonably expect to remove that way, but the PCB will still not move. You now need to ask a friend for help, preferably a nicotine addict. Grip the PCB carefully with wide pliers, hold the lock firm with a second set of pliers or a gloved hand and carefully use a lighter to heat the cylinder around the PCB. Apply gentle pulling power. At some point (hopefully before it smells too bad) you will feel the PCB move out of the cylinder. Be aware that the cylinder is hot after this procedure, so don´t touch with bare fingers for a moment.

After bit of further scratching and scraping, the chip side of the PCB from the cylinder looks like this (enlarged):



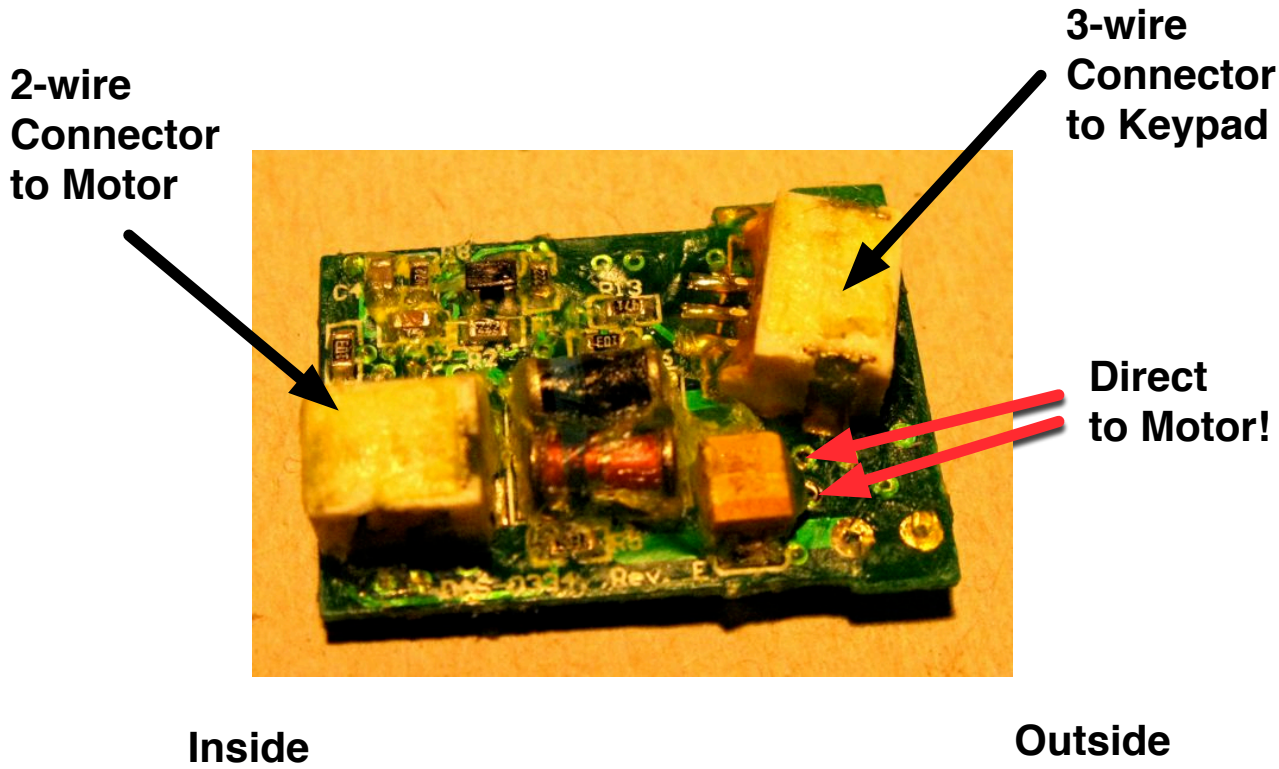
The big chip is the MASK200501 that we know already from the keypad. The smaler 8-pin to the right of it is a Microchip 24AA02 2kbit Serial EEPROM. Since there are other versions of the TSE 3000 lock that can store more unlock codes the EEPROM is most likely used for this kind of config data. It may be reasonable assume that they use the same cylinder for the other versions and just vary the keypad.

There are a number of interesting looking testpads, that are fascinatingly near to the edge of the PCB that is facing to the outside. Due to the residues of the ectoplasma jelly tracing the testpads is a bit challenging.

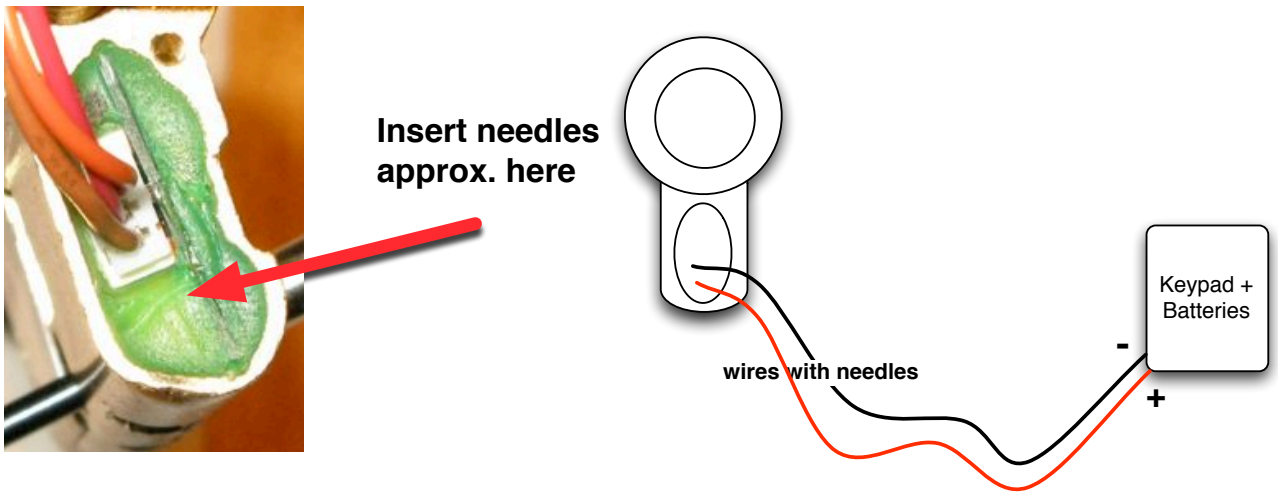
We did not bother to analyse the MASK200501 microchips or the EEPROMs in detais, as the effort necessary was not justified due to the low resistance to easier attacks.

## The Bypass to the Motor

The 5000-Dollar question is of course if there are points on the cylinder PCB that can be reached from the from front of the cylinder to drive the motor directly. A quick tour with a multimeter across the cylinder PCB reveals two contact pads that can be reached from the front side of the lock that provide a direct path to the motor.




To bypass the whole code system, an attacker just needs to stick in two needles from the front of the cylinder through the green goo until they hit the orange capacitor that sits behind the contact points. Then scratch towards the PCB, apply 3-5V of power and wiggle a bit till you get contact. If the motor turns into the wrong direction, reverse polarity. The voltage to turn the motor can be easily obtained from the battery pack in the keypad. If you want it more comfortable, you can heat the needles a bit. Note that you need to make contact to both points, since the motor driver electronics isolates the motor from common GND in order to drive it forward and backward. Two fixing pins pushed in parallel 2 millimeters apart through the cork from a wine bottle do the job nicely. If you don't like working blind and don't care about leaving traces, simply scrape away the goo up to the capacitor.



# Conclusion & Recommendations

The Burgwächter TSE 3000 Basic Code provides security only against attackers who have absolutely no clue at all about electronics. We identified two viable and easy attacks that are not prevented by the locks architecture and construction (and there are certainly more...).

If you have such a lock in use and can not easily replace it, you should:

1. put a real drill protection shield made from hardened steel in front of the cylinder to prevent access to the cylinder PCB from the front. The lock cover shipped with the lock is inadequate and by far not solid enough to provide this kind of protection. Be aware that the cable routing from the lock to the cylinder might be affected.

2. make sure that there is no way to covertly place a microchip onto the wire between keypad and cylinder for the replay-attack. This is really hard, since the keypad housing provides plenty of space and its enclosure is easy to remove to replace the batteries. Maybe you can come up with a way to put the keypad into a steel enclosure that is secured by an mechanic lock?

The manufacturer should:

1. as a stop-gap measure switch to a real drill protection enclosure. The needle-attack is trivial and devastatingly effective, if practiced a bit.

2. rethink the design of the cylinder PCB to have no contact points going directly to the motor reachable with needles from the front.

3. change the wire protocol between keypad and cylinder to something with proper encryption and salting of the transmissions, so simple replay will not work anymore.

4. Think of a way to prevent or signal opening of the keypad enclosure to the user, so inserting the microchip for replay becomes harder.

All these measures would not make the Burgwächter TSE 3000 a high security lock, but at least make it harder for an attacker (and give us something more interesting to play with again).