Karacell Cracking Competition

http://karacell.info
Document version 2
May 3, 2013

0. Background

The website on the title page contains everything you need to participate, including email addresses to which to mail questions or solution submissions.

Tigerspike is proud to provide source code to the Karacell 3 ("Karacell") cryptosystem to the open source community. We realize that good security requires openness. However, openness is not sufficient to prove security. Therefore, we are providing the additional incentive of prize money to encourage people to find holes in the cryptosystem. Ultimately, it must be said that no cryptosystem has any particular level of proven strength in a mathematical sense. But more hackers with access to the code can only help.

To that end, we encourage the community to form networks of machines in order to attack the problem collaboratively. We realize that the value of our algo is a monotonically increasing function of prize money and time uncracked. Thus our strategy (and frankly, the only strategy that would fly with the accountants) is to start small and hopefully increase the prize amount over time, until it is claimed.

On the website, we have posted encrypted files under “Contest Files”. One set of files is an example, which appears decrypted (example.bin), encrypted without a hash (example_hash_none.kcl), encrypted with an LMD7 hash (example_hash_lmd7.kcl), and encrypted with an LMD8 hash (example_hash_lmd8.kcl). You can download the source code from the website as well, and use it to practice decrypting the example files with the key in key.txt.

The contest files are analogous to the example files in that they all equate to the same decrypted contents (but not the same as the example) and have the same various hashes.

Specifically, the decrypted contents are 32 KiB in size, only the first 256 bits of which being nonzero. Thus you have 8 blocks of 4 KiB each, which should provide ample amounts of xor masks to allow you to derive the xor mask fragment which was applied to the first 256

bits, and thereby effect decryption.

For the sake of verification, the first 128 bits are repeated verbatim, forming the first 256 bits. It is the first 128 bits, in big endian (most significant bit first) form, that you must provide in order to have an opportunity to win the prize, as explained in the next section. If possible, we suggest trying your cracking methods on the example files first, in order to verify your approach.

1. Rules

You may use any computer or network thereof, whether analog, digital, or quantum. Any computer language is also acceptable.

1.1 Prize 0

Prize 0 begins at USD5000. This prize shall be awarded to the sender of the first email we receive which contains the first 128 correctly decrypted bits of contest_hash_lmd7.kcl (see “Contest Files” on the website) in big endian order, and sends us an English explanation of the method used to find it. (The other contest files contain the same decrypted contents, but we mention this one in particular because the hash can be used to verify correctness.)

Once a winner of Prize 1 has been identified, this prize will no longer be awarded.

As to the required explanation, our goal is not to prevent losing a few thousand dollars. To the contrary, we would be happy to pay such a fee in order to find, and hopefully correct, defects in our implementation or the underlying algo. In fact, the accountants must assume that we will lose the money, which is one reason that we need to start small. We think of it as a consulting fee for hiring a security expert who will have found a problem with our software.

We will not unduly withhold prize money because your explanation is grammatically incorrect or complicated. It should merely be sufficient

to allow us to reproduce your work, given adequate time, money, and expertise. The bottom line is that we have already "paid" the prize money from an accounting perspective. All we want is the explanation of weakness that we paid for. If we can't fix the weakness, that's our problem, not yours.

If you think you've cracked the code, then please verify that the first and second 128 bits are equal -- just like in the example files -- before emailing the address posted on the website. If you prefer another delivery method, then email us with your request and we will try to work with you.

1.1 Prize 1

Prize 1 begins at USD10,000. This prize shall be awarded to the sender of the first email we receive which contains the hexadecimal key used to encrypt contest_hash_lmd7.kcl, and sends us an explanation of the method used to find it, of the same standard employed for Prize 0.

Once a winner of Prize 0 has been identified, this prize will no longer be awarded.

The key has a numerical value between 2^120 and 2^121, which is the weakest key that Karacell supports. For example, 1768AE41868F78F231888B3717231FCC2 would be a plausible solution.

Given that contest_hash_lmd7.kcl contains an LMD7 hash, it will be easy for you to verify your proposed solution. So please first try decrypting the file from the command line using the compiled Karacell source code available on the website. If decryption silently succeeds in producing a 32 KiB file, then your solution is probably correct. But for further assurance, please verify that the first and second 128 bits of the decrypted file are identical and nonzero.

1.2 Prize Payment

We cannot promise to support payment modes other than certified check or wire transfer.

If you prefer some other payment method, we may or may not be able to accomodate you. So, once you have confirmed the decryption of a repeated 128-bit code, then please email us with your suggested arrangement *before* you send us your solution.

2. Other Results of Interest

While we only offer prizes of the above description, we would be happy to hear from you if you can show that our security assumptions about Karacell are materially overestimated. In particular:

(1) Trying every key is grossly unnecessary. One need only try some small fraction thereof, for example, the square root as many. (But if you find a way to save 50% of the cracking time, that's not really interesting from a security perspective.)

(2) Karacell files have a bitlane bias somewhere, i.e. apart from their length, they are somehow statistically discernible from binary noise. (Obviously this question cannot be answered with just a handful of example files.)

(3) Xor mask reuse occurs much more often than the period of the embedded Marsaglia oscillator would suggest.

(4) The source code contains memory leaks or other bugs, whether or not exploitable.

(5) A quantum algo exists which could quickly crack Karacell with fewer than 10K qubits. We realize that the Karacell Table could be stored in as little as 2^8 qubits. But then no single universe would have access to the entire table; only a classical observer could browse it in its entirety. There might be some way to partition the table and access it via superpositional tumblers, to computational advantage

over classical methods. However, such methods would appear to preclude the global sorting of partial sums, which is instrumental to Sparse Horowitz and Sahni.

3. FAQ

Q. What if I decrypt the file and see the 128 repeated bits, but Tigerspike says my solution is wrong? Who's to say who's right?

A. We have the key to the contest files on third-party cloud storage. (Yes, it's encrypted with another key.) This key will allow us to inspect the decrypted contest files and verify your answer. If you still don't trust us, then we recommend that you try for Prize 1, which allows you to test your solution automatically, simply by decrypting the file and not getting an error message about a failed hash. Third parties could then easily verify the same.

Q. What if I manage to decrypt the file correctly, but it's because I exploited a bug in the implementation? In other words, I didn't find a problem with the algo itself. Am I still eligible for the prizes?

A. Yes! You need only tell us which bug you exploited. Fixing it is our problem, not yours.

Q. What if I use the hashes to crack the file, instead of attacking the xor mask itself? Am I still eligible for the prizes?

A. Yes again! But that might be difficult, considering that the hashes are also encrypted.

Q. Can I hack the source code or the OS to say that the file decrypts correctly, when that's not actually true?

A. Sure. But you can only win a prize by being the first to provide the 128 bits mentioned above, or the 121-bit key.

Q. I know how to crack this, but I would need more than the prize

money to do it. So can you increase the prize money?

A. Let's just say that this will be an ongoing feedback process with the accountants. In the meantime, of course, you can mouth off online about why it's weak.

Q. What if you receive multiple correct solutions?

A. We'll take the one that arrived first. There will be only one winner.