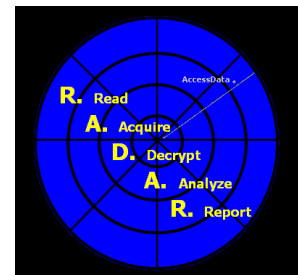




# PORT

**Portable Office<sup>®</sup> Rainbow Tables  
Sales and Promotional Summary**



ACCESSDATA, ON YOUR RADAR

## ***What are Rainbow Tables?***

Rainbow tables are a set of pre-computed lists used to significantly accelerate a brute-force attack on specific files. A brute-force attack is a process of deriving a cryptographic key by trying every possible combination within a specific keyspace until the correct one is found.

## ***How do Rainbow Tables Work?***

Having a list of keys in which to find one particular key is a faster method than simply guessing random keys until you find the one you want. The keys in a rainbow table are random, so rainbow tables cannot cover 100% of the keyspace. The coverage is probabilistic.

For example: if you rolled a dice six times, you probably wouldn't get each number (one through six) each time. You'd more likely miss some numbers, while getting duplicate occurrences of others. Imagine instead a trillion-sided die. You could roll it, say, ten trillion times and still not get every last number at least once. Because of the sheer amount of rolls and possibilities, however, you would get over 99% of each number at least once.

The enormous list of keys and the search through it take huge amounts of memory. You can reduce the size of the lists by storing a fraction of them in an abbreviated form that can be reconstructed when needed.

## **Chaining**

Creating this abbreviated form is called chaining. A chain is a range of possible keys and ciphertext created from one given key. Chains are created by choosing a key, deriving ciphertext from it, and then deriving a new key from that ciphertext. This new key is then used to make new ciphertext, and so on for however long you want to make the chain. The chain cannot be generated in reverse, that is, you can't decrypt the keys.

You then abbreviate the chain by removing all but the first and the last keys to save space. Later on, the middle of the chain can be reconstructed from the stored points. This extra step (reconstruction) requires additional time, so you're faced with a performance trade-off: less memory needed to store the collection of chains (the rainbow table), but more time needed to compare the key to the collection of chains.

## **Time/Memory Trade-off**

You can store more keys by making longer chains, and increase your probability of guessing. The longer your chains are, on the other hand, the more time it takes to reconstruct them when you are processing a document.

## ***What is AccessData's Portable Office® Rainbow Table?***

AccessData's Portable Office® Rainbow Table provides a collection of decryption key chains, and the mechanism to move them to your Microsoft® Office files.

The PORT interface processes your Office files one job, or file, at a time, and in the order in which they appear on the main window. It provides information about each job's progress in a Statistics window.

AccessData's PORT provides folders to organize the files on which you're working. During setup, you specify folders to manage Input files, Output files, and logs. You then move the files you want decrypted to your Input folder, and wait for PORT to process them and move them to the Output folder. You can view an event log when your files are finished processing.

With PORT's application of the Time/Memory trade-off, you can now take a huge list of successful decryption keys to the jobsite. Compare the size of PORT to our regular rainbow tables:

	Hash Table	PORT
Accuracy	100 %	99 %
Storage Size	8.1 TB	4 GB
Speed per File	30-90 seconds	5 + minutes
Applications	Office, PDF, LAN	Word and Excel

PORT is much smaller and more manageable, with accuracy reduced by only one percent.

## **Contact Us**

Sales  
384 South 400 West  
Suite 200  
Lindon, UT 84042  
USA

[sales@accessdata.com](mailto:sales@accessdata.com)

**800.574.5199**