Crypto/Math Paper
SAND2019-14009R

Effect of Partial Key Knowledge

W.R. Cordwell

Sandia National Laboratories

November 2019

U.S. DEPARTMENT OF ENERGY NNSA

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S.Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories

Introduction

For strong cryptologic algorithms, it is often assumed that exhaustive search (AKA "brute force") will take 2^b trials, where b is the number of bits of the secret key. What happens, though, if an adversary gains partial knowledge of the secret key? Perhaps he has intercepted a garbled transmission of the key, where he knows the maximum number of garbles, but not where they occur, or perhaps he knows the probability of each bit being correct. How much does this help him?

First Model–Garbled Bits

In the first case, assume that an attacker knows 256 (but garbled) bits of a key. Suppose that the attacker knows that there are at most n bits that are incorrect, but he has no idea which bits are correct and which are wrong. He tries the key that he received, and it doesn't work. Knowing that it is garbled, he tries flipping a single bit in each of the 256 bit positions. It still doesn't work, so he tries flipping all possible pairs of two bits. He keeps proceeding, hoping that he will get the correct key. How do we estimate how much work it will take the adversary to find the key?

In this case, the approach to calculating the number of tries required by the attacker is to add up the binomial coefficients, or the number of ways one can choose j bits out of 256, up to the maximum unknown number, n. This is $N = \sum_{j=0}^{n} \binom{256}{j}^{\dagger}$. *Assuming that each of these tries is equally likely to succeed*, we have the probability of success for each try of $p = \frac{1}{N}$. The average number of tries is then $\langle N_{tries} \rangle = \sum_{j=1}^{N} j \cdot p = \frac{N(N+1)}{2} \cdot \frac{1}{N} = \frac{N+1}{2}$,

[†]note that, if $n = 256$, we are guessing all possible choices, and this sums to 2^{256}

if there are at most n garbled bits in the key. The factor of $\frac{1}{2}$ is because, on average, the adversary would need to guess about half of the possibilities before he guesses correctly.

Figure 1 shows the effective key strength (in bits, so it is the base 2 log of the number of tries) as a function of the maximum number of bits that an adversary needs to change.

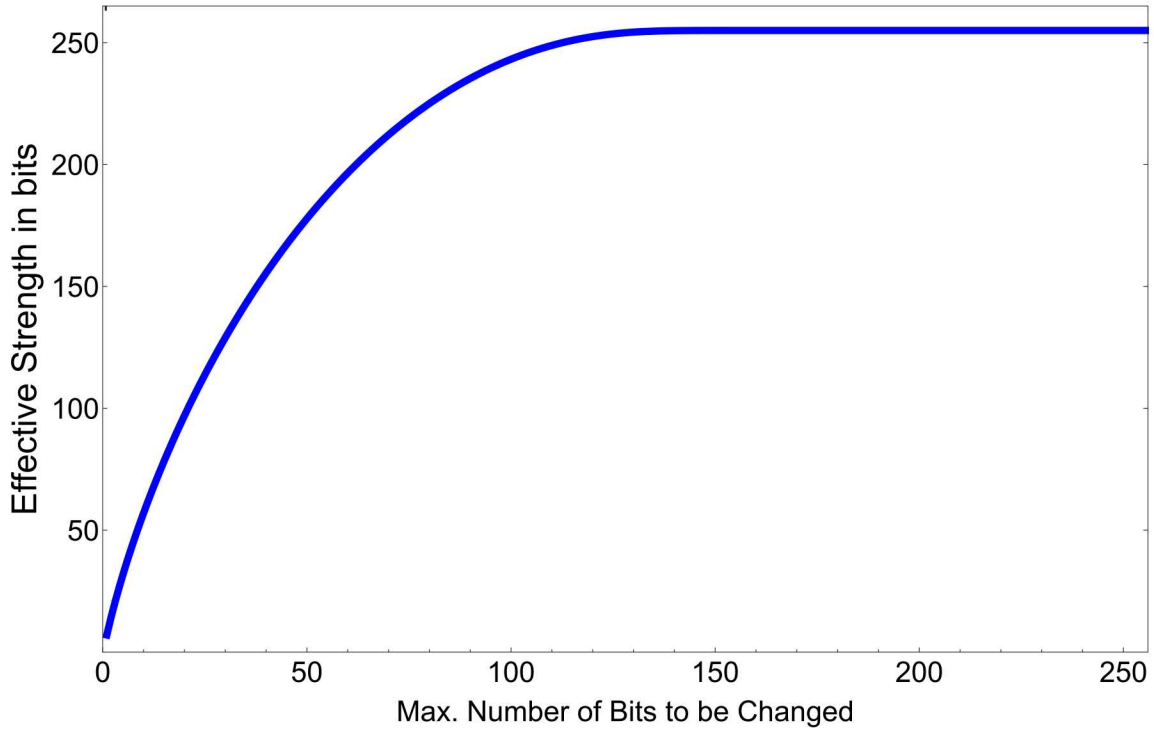

Figure 1: Remaining Strength of Key as a function of required num. of guesses

This analysis assumes that each arrangement of bits has the same probability of being correct. That is, for example, the case where a single bit, bit number 37, was changed had the same likelihood of being the correct key as if five bits, numbers 8, 23, 24, 149, and 200, were changed.

Explicit values for a few values of n are listed in the table below.

Table 1	
Max. num of unknown bits	Effective Key Strength (bits)
160	255.000
144	254.972
128	254.070
112	249.748
96	240.317
80	225.019
64	203.141
48	173.633
32	134.639
16	82.158

Second Model–Probabilistic Bits

It might be the case that the adversary is unlikely to know that "at most n bits are incorrect" without knowing anything more.

Suppose, instead, that we have a case where the attacker knows each bit to some probability. We can then again calculate the expected number of guesses required to find the correct key.

Let $n = 256$ be the number of bits, so that $2^n = 2^{256}$ is the total number of possible keys. The formula for the expected number of guesses is $\langle N \rangle = \sum_{i=0}^{2^n} f(i) \cdot p(i)$, where $p(i)$ is

the probability that the i^{th} guess is correct, and $f(i)$ is the number of guesses made up to that point (including the guess for the i^{th} case).

For example, suppose that the adversary knows each bit with only $p = \frac{1}{2}$; that is, he knows nothing about the real key. Then each guess has equal probability, $(\frac{1}{2})^{256}$, and the expected number of guesses is just $\sum_{i=1}^{2^n} i \cdot (\frac{1}{2})^{256} = \frac{n(n+1)}{2} = \frac{n+1}{2}$, which is almost exactly $\frac{2^{256}}{2} = 2^{255} = 2^{n-1}$. This is why, as noted above, people usually say that exhaustive search takes half the total number of guesses, on average, before the correct key is found.

Now, suppose that an adversary knows the bits with a probability for each bit of $p > \frac{1}{2}$. Let $q = 1 - p$, so q is the probability that the bit is not guessed correctly. Clearly, it is to the attacker's advantage to try the most likely possibilities first. In this case, he tries the original guess, then he varies one bit at a time. If that does not work, then he varies pairs of bits, etc. The first try (try number 1) has $P = p^n$ chance of being correct. The next $256 = \binom{256}{1}$ tries, tries of numbers 2 to 257, occur each with probability $p^{n-1}q^1$. Varying two bits at a time, tries of numbers 258 up to $\binom{n}{0} + \binom{n}{1} + \binom{n}{2}$ occur each with probability $p^{n-2}q^2$, etc.

The results are plotted in Figure 2. The effective key strength tapers off slowly until $p \approx 0.8$, where it starts dropping steeply, and very steeply after $p \approx 0.9$. For an effective strength of 192 bits (out of an initial 256 bits), one would not want an adversary to have better knowledge than about $p = .85$, or 85%. Numerical results are listed in Table 2.

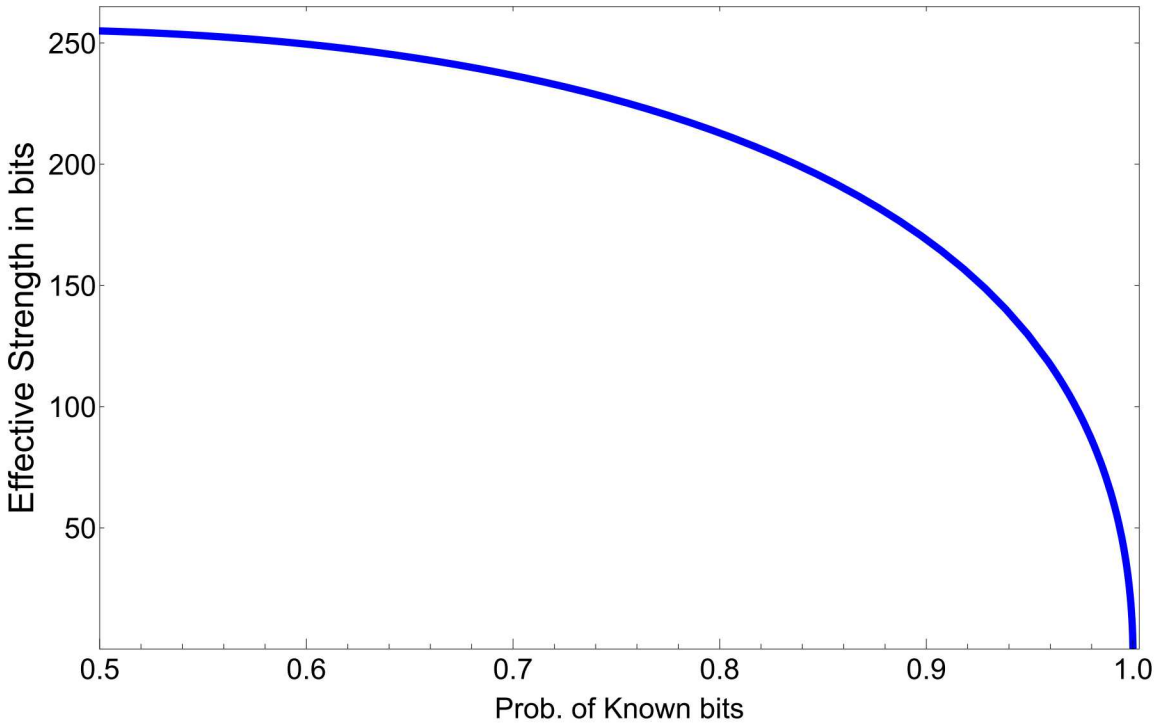

Figure 2: Remaining Strength of Key as a function of prob. of knowing each bit

Table 2: Bit Probability Knowledge/Effective Key Strength (bits)									
0.50	255.000	0.60	249.537	0.70	236.66	0.80	212.907	0.90	168.957
0.51	254.716	0.61	248.624	0.71	234.855	0.81	209.684	0.91	162.483
0.52	254.381	0.62	247.636	0.72	232.941	0.82	206.261	0.92	155.385
0.53	253.992	0.63	246.568	0.73	230.911	0.83	202.623	0.93	147.537
0.54	253.546	0.64	245.42	0.74	228.76	0.84	198.749	0.94	138.765
0.55	253.041	0.65	244.188	0.75	226.483	0.85	194.618	0.95	128.819
0.56	252.474	0.66	242.869	0.76	224.071	0.86	190.202	0.96	117.311
0.57	251.842	0.67	241.46	0.77	221.516	0.87	185.471	0.97	103.585
0.58	251.144	0.68	239.958	0.78	218.811	0.88	180.385	0.98	86.3546
0.59	250.376	0.69	238.359	0.79	215.945	0.89	174.900	0.99	62.2868

A Slightly Different Viewpoint

We have been considering the expected number of guesses as a measure of the strength of the system. This is, roughly speaking, akin to considering the Shannon Entropy involved. When an adversary might have more than one part to exploit, he might, for example, be content to try guessing until he has spanned more than half of the total probability involved for a part. If that did not work, he might try a different part. This is more akin to considering the min-Entropy of the part.

Since, for high values of p, the most likely values are in the first portions of the search, this means that the adversary might not need to try as much as one might otherwise think.

Given an estimate of the adversary's resources, we can predict the point at which his search is infeasible. For example, if an adversary can perform about 2^{60} guesses, then he can "win" if the bit probability exceeds $p \sim .96$, whereas the table above says that the exhaustive search for $p = .96$ has about 117 bits of security. For the ability to perform 2^{80} guesses, the bit probability cutoff is $p \sim .94$.

Different Knowledge of Different Sections

If the adversary has quantitatively different knowledge about different portions of the key, the analysis is somewhat more complicated. As an example of why, and why this is important, consider where the adversary knows all of the 256 bits with probability 0.75, vs. the case where he knows half of the bits perfectly ($p = 1.0$) and half not at all ($p = 0.5$). In both cases, the expected number of bits that he knows is three quarters of all of the bits, but in the first case, he must do an amount of work equivalent to more than 2^{226} trials, while in the latter case, he need only do an amount of work equivalent to 2^{127}. Knowing a lot about a portion of the key is very valuable to the attacker!

In order to outline the method for computing the expected amount of work, we consider a case where the adversary knows half of the bits with a probability $p_1 = .90$, and the other half with probability $p_2 = .70$.

The optimal strategy is for the adversary to first flip bits where he has the most uncertainty. At some point, though, it gives a better total probability to flip one of the higher probability bits instead of several of the lower probability ones. The ordering is not obvious; for our case above, the first few choices, from highest probability to lowest, are

> no bits flipped, 1 p_2 bit flipped, 2 p_2 bits, 1 p_1 bit, 3 p_2 bits, 1 p_1 and 1 p_2 bits, 4 p_2 bits, 1 p_1 bit and 2 p_2 bits, ...

When the expected number of tries is computed, it gives an effective strength of 202.52 bits.

Table 3 gives a listing for several choices of p_1 and p_2, each applying to separate 128-bit sections (256 bits, total), all with average probability of 0.80. The first case, where $p_1 \approx p_2$, gives the same result as if all of the bits were known with probability 0.80 (as it should). The last case, where half the bits are known almost certainly and half with just above 0.60 probability, gives the same result as for testing 128 bits with a probability of 0.60, agreeing with the answer produced by the direct method for 128 bits.

We see that, even if the "average" knowledge is constant, as knowledge of one area becomes more precise, even though the knowledge of the other area becomes correspondingly less precise, the search requirement drops dramatically.

Table 3: Bit Probability Knowledge/ Effective Key Strength (bits)	
$p_1 = 0.79999999$ $p_2 = 0.80000001$	212.907
$p_1 = 0.75$ $p_2 = 0.85$	210.435
$p_1 = 0.70$ $p_2 = 0.90$	202.520
$p_1 = 0.65$ $p_2 = .95$	185.919
$p_1 = 0.600000001$ $p_2 = 0.999999999$	123.779

It is straightforward to extend to more general cases, with three or more probabilities and different-sized areas to which they apply. The number of terms that require sorting starts getting large, but the computation should be feasible for a few to several different probability values.

Summary

A graph and table are given for the model where bits are garbled up to a maximum number of bits. For the probabilistic case, the effective key strength as a function of an adversary's knowledge (single probability for all bits) is give in Table 2. For an effective key strength of around 192 bits, the adversary should have at most a probability of .85 of knowing each bit.

If the adversary knows more about one section than another, this can be a major benefit to him, and this gives a much weaker effective key strength than does using the average probability. This is illustrated in Table 3.