Privacy Amplification

Ivan Damgård
(Note for Cryptography Course)

September 17, 1998

1 Motivation

Suppose you are storing on your computer a number of keys to conventional cryptosystems such DES. Such keys usually have the form of randomly selected binary strings. Suppose you find out that a hacker has broken into your computer and has been able to find and access some, but not all of your keys. This could be a reasonable assumption if you know that he was detected and stopped relatively quickly after the break-in occurred, and so he would not have had time to actually locate and retrieve all your keys. The problem, however, is that you have no idea which of the keys he actually got hold of.

Thus it would not be secure to go on using any of the old keys, since any single one could now be known to the adversary. So is there a way to recover from this situation, i.e. is there a way to create from this partially compromised data some key material that you can safely use?

The answer is yes, and the method to use is privacy amplification using universal hash functions. This can be described as a general protoocl for two players A and B who start from a shared binary string w, about which an adversary, E knows some partial information (to be made precise below). The idea will be that for example A will choose at random a hash function h from a class of so-called universal hash functions $\mathcal{G}$, and will send this to B. Then they both replace w in their memory by $h(w)$. We will show below that if the class of hash functions is correctly chosen, then the expected amount of information that E has about $h(w)$ can made arbitrarily small. This will hold *even if E knows the hash function h*. Thus it is sufficient if A can send h to B over a channel where E can listen to all messages sent, but cannot modify them [1].

Using the outlined method in the partial break-in example above requires of course that you are communicating with some other party who shares all the keys with you (otherwise he will not be able to do the hashing on his side), but this can actually occur in practice, e.g. if you are doing communication with a bank or another central data processing facility. It is very common here to have several keys for different purposes, e.g. different kinds of data, even though they are all used for communication between the same two parties.

There is an almost infinite list of other scenarios where universal hashing comes in handy, e.g. probabilistic proof systems, randomized algorithms, all kinds of cryptographic

[1] Even the assumption about the existence of this channel can be removed, but we will not go into this here

protocols including the sattelite scenario, etc. But that is another story..

2 Information Measures

The first thing we need to do is to make precise what we mean by E having "partial information" on w. Should we say that she knows at most t of the physical bits of w? This may suffice for the example in the previous section, but is not at all general enough for many other applications. For example, it might be the case that E knows $f(w)$ for some function f that outputs t bits (but where we have no idea which function it is). Even less tangible, perhaps there are some of the bits of w that E can guess with good probability. In this case there no piece of information on w that she is sure of, but still she "knows something".

The most general definition we can come up with would be to use Shannon's entropy concept and say the following: Let us think about w as a possible value of a random variable W: if w is an n bit string, then the maxmimal uncertainty E can have about W is n bits. Now, if we let random variable V denote the total collection of what E knows, then $H(W|V)$ measures her uncertainty about W, and so $n - H(W|V)$ measures the amount of information V gives her about W.

However, this turns out to be too general. If we allow E to learn information about W in any way that obeys a restriction of form $n - H(W|V) \leq t$, we will run into cases, where no method for compressing w will work, as the following shows:

Exercise

Suppose V is a variable that is equal to W with probability 1/2, and otherwise has a special value ?. Show that for any $r \leq n$ and any function $g : \{0,1\}^n \rightarrow \{0,1\}^r$, we have $H(g(W)|V, g) = r/2$

In other words there are cases where if E knows "half of w" using Shannon's information measure, then no matter how we compress, she knows half of the output. We therefore need to make a stronger assumption about the information E knows. The solution turns out to be the so called Rényi entropy. For random variable X, we dcefine the *collision probability* of X:

$$P_c(X) = \sum_x P(X = x)^2$$

i.e. the probability that two independent outcomes of X yields the same result. Then the *Rényi entropy* $R(X)$ is defined as:

$$R(X) = -\log_2(P_c(X))$$

Like $H(X)$, $R(X)$ measures the "flatness" of the distribution of X. However, it is in some respects not as nicely behaved as Shannon's entropy, and so one usually prefers Shannon's measure. But in this particular case, $R(X)$ is just what we need. We define conditional entropy in the usual way:

$$R(X|Y) = \sum_y P(Y = y) R(X|Y = y)$$

Exercise

Use Jensens inequality to show that $R(X) \leq H(X)$

3 Universal Hash Functions

We first make the following definition:

A class $\mathcal{G}$ of functions $\mathcal{A} \to \mathcal{B}$ is called *universal* if for any distinct $x_1, x_2 \in \mathcal{A}$, the probability that $P(g(x_1) = g(x_2)) \leq 1/|\mathcal{B}|$, when g is chosen randomly in $\mathcal{G}$.

One can also define strongly universal hash functions, where one requires equality in the above, or that the images $h(x_1), h(x_2)$ are independent when h is randomly chosen. In any case, we are after a class of functions that emulate some essential properties of random functions, but such that the description of a function in the class has managable size.

An example is the following: let $\mathcal{A} = GF(2^n)$, $\mathcal{B} = GF(2^r)$. Then let $\mathcal{G}$ be the class of functions that contains a function g_a for each $a \in GF(2^n)$, such that $g_a(x)$ is the first r bits of ax, where interpret the elements in $GF(2^n)$ as n-bit strings in the natural way. This class is universal (why?).

4 Results

We first get the following

Theorem

Let X be a random variable taking values in set $\mathcal{X}$, and let G be the random variable corresponding to choosing a random function in universal class $\mathcal{G}$, of functions from $\mathcal{X}$ to $\{0,1\}^r$. Then

$$H(G(X)|G) \geq R(G(X)|G) \geq r - \log_2(1 + 2^{r-R(X)}) \geq r - \frac{2^{r-R(X)}}{ln2}$$

This says that if $R(X) >> r$, then even if I tell you what function in g I use to compress a value x of X with, your uncertain about $g(x)$ will almost maximal, this will hold on average over the choice of x and g.

To prove this, first note that the first inequality follows from the exercise above. Then we compute as follows:

$$\begin{aligned}
R(G(X)|G) &= \sum_g P(G=g)R(G(X)|G=g) \\
&= \sum_g P(G=g)(-\log_2 P_c(G(X)|G=g)) \\
&\geq -log_2(\sum_g P(G=g)P_c(G(X)|G=g))
\end{aligned}$$

where the last step follows from Jensens inequality. The sum inside the logarithm is equal to the probability that when we choose g at random, and independently of this, choose

x_1, x_2, then $g(x_1) = g(x_2)$. This probability can also computed as follows, letting X_i denote the choice of x_i:

$$\begin{aligned} P(G(X_1) = G(X_2)) &= P(X_1 = X_2) + P(X_1 \neq X_2)P(G(X_1) = G(X_2)|X_1 \neq X_2) \\ &\leq P_c(X) + (1 - P_c(X))2^{-r} \\ &< 2^{-R(X)} + 2^{-r} \\ &= 2^{-r}(1 + 2^{r-R(X)}) \end{aligned}$$

Inserting this in the above gives the result immediately, if we also note that in general $\log_2(1+y) \leq y/ln2$.

We get immedately the following

Corollary

Consider any joint distribution of V, W, where $V = v$ is a particular piece of information observed by E and A, B know the value of W. If we know that $R(W|V = v) \geq c$, and we let A, B choose $K = G(W)$ as a secret key, where G is defined as above, then using the above theorem where we let X have the distribution of W when $V = v$ is given, we get

$$H(K|G, V = v) \geq r - \frac{2^{r-c}}{ln2}$$

So when $r - c$ is large, E's uncertainty about K is almost maximal.

Finally, we have a corollary that pertains directly to the partial break-in example above, namely when we assume that E knows some function of W outputting at most some number of bits, but we do not know which function is involved:

Corollary

Suppose W is a uniform n-bit string with uniform distribution, let $V = e(W)$ for an arbitrary eavesdropping function $e : \{0,1\}^n \rightarrow \{0,1\}^t$, where $t < n$. Let $s < n - t$ be a positive security parameter, and let $r = n - t - s$. If A, B choose $K = G(W)$ as their new secret key, G defined as above, then E's expected uncertainty about K, given V and G satisfies

$$H(K|GV) \geq r - \frac{2^{-s}}{ln2}$$

To prove this, let c_v be the number of $w \in \{0,1\}^n$ that satisfy $e(w) = v$. From E's point of view, if she knows that e applied to A and B's value of W is some v, then there are c_v equally likely candidates. Thus

$$P_c(W|V = v) = c_v \cdot (1/c_v)^2 = 1/c_v$$

and so $R(W|V = v) = \log_2 c_v$. So we get directly from the corollary above that

$$H(K|G, V = v) \geq r - \frac{2^r}{c_v \cdot ln2}$$

We can now compute an estimate of $H(K|GV)$ from the estimate of $H(K|G, V = v)$ by averaging over the value of V, using that $P(V = v) = c_v 2^{-n}$.

These results were taken from Bennett, Brassard, Crépeau and Maurer: *Generalized Privacy Amplification*, IEEE Trans.Info.Theory, vol 41 (1995) pp.1915-1923. Here, more results on this subject can also be found.