Detailed analysis of parameter values in BC(BSC)

Adam Smith

June 1997

1 The basics

1.1 Notation

Table 1: Notation used

ϵ	BSC error probability
n	Length of code
k	Dimension of code
d	min. distance of code
R	Transmission rate ($= r/n$)
γ	Relative bound on dimension of code
λ	Criterion for accepting/rejecting Alice's codeword
$h(x)$	Entropy function (base 2)
C_ϵ	Capacity of a BS_ϵ
δ	$R = C_\epsilon + \delta$
s	Privacy Amplification security parameter
r	Privacy Amplification final message length
t	Number of bits of the syndrome
β	PA param: $s = n - r - t - nC_{\epsilon+\beta}$
p_h	Probability that honest Alice fails Bob's test
p_m	Prob. that random systematic code has $d < \gamma\epsilon$
p_a	Prob. that Alice can cheat
α	Average amount of info. Bob is allowed

1.2 Background

Theorem 1 (Bernstein, in Renyi61) *If $X_1, X_2, \ldots, X_n$ are independent random variables such that $D = \sqrt{\sum_i V(X_i)}$, $E(X_i) = 0$ and $|X_i| < K$ for all i, and if $\mu \leq D/K$, then*

$$Pr\left[\sum_i X_i \geq \mu D\right] \leq e^{\frac{-\mu^2}{2(1+\mu K/2D)}}.$$

Corollary 2 *Let $p \in (0, 1/2)$, let $q = 1 - p > 1/2$. If $X_1, X_2, \ldots, X_n$ are independent random Bernouilli variables of probability p or q (not necessarily all identical) such that $\mathcal{X}$ is their expected sum, and $\lambda < p$, then*

$$\begin{aligned} Pr\left[\sum_i X_i - \mathcal{X} \geq \lambda n\right] &\leq e^{\frac{-n\lambda^2}{2pq(1+\lambda/2p)}} \\ Pr\left[\sum_i X_i - \mathcal{X} \leq -\lambda n\right] &\leq e^{\frac{-n\lambda^2}{2pq(1+\lambda/2p)}} \end{aligned}$$

Lemma 3 (MS77) *If $\mu \in (0, 1/2)$ and $n \in N$, then*

$$\frac{2^{nh(\mu)}}{\sqrt{8n\mu(1-\mu)}} \leq \binom{n}{\mu n} \leq 2^{nh(\mu)}$$

2 Analysis of different error probabilities

2.1 Finding a good code

We need to find a code such that both k and d are high. We don't need to be able to decode received codewords, only to check if a given word is part of the code and to generate codewords efficiently. For this we will use systematic linear codes, i.e. codes with a generator matrix of the form $(I_k|Y)$ where Y is a random $k \times (n-k)$ boolean matrix.

Theorem 4 (Based on MS77) *Let $n, k \in N$ such that $k < n$ and $R = k/n$. Let $S_0 = \{$code generated by $G = (I_k|Y)|Y \in M_{k\times(n-k)}(F_2)\}$. If $R < C_\delta$ then the probability of choosing a code at random in S_0 with min. distance smaller or equal than δn is at most $2^{n(R-C_\delta)}$.*

Proof: For a given n-bit word w, define

$$\begin{aligned} N(w) &= \#\{C \in S_0 | w \in C\} \\ &= \#\{Y \in M_{k\times(n-k)}(F_2) | w_{1,\ldots,k}Y = w_{k+1,\ldots,n}\} \end{aligned}$$

where $w_{i,\ldots,j}$ refers to the word formed by bits i through j of w.

Clearly, if w is the null vector, then $N(w) = 2^{k(n-k)}$. Also, if $w_{1,\ldots,k}$ is null but $w_{k+1,\ldots,n}$ is not, then $N(w) = 0$. So let us now investigate the case where $w_{1,\ldots,k}$ is non-zero.

For a given matrix Y, each bit w_{k+i} of $w_{k+1,\ldots,n}$ can be seen as a parity check over certain bits of $w_{1,\ldots,k}$. The exact parity check is determined by the entries in the corresponding column v_i of Y. For a given bit of w_{k+i}, how many different (n-k)-bit vectors are possible for v_i?

Let us suppose that $w_{1,\ldots,k}$ has j zeros and $k-j$ ones. Then there are j positions in v_i which don't matter at all (they can hence be set in 2^j different ways). Also, the total parity of the $k-j$ other bits must equal to w_{k+i}. There are exactly 2^{k-j-1} ways to set $k-j$ bits so they have a given parity. The total number of possiblities for v_i is thus 2^{k-1}. Because there are $n-k$ different vectors in Y, there are a total of $2^{(k-1)(n-k)}$ matrices Y such that $w_{1,\ldots,k}Y = w_{k+1,\ldots,n}$. Hence, $N(w)$ is constant for all vectors w with at least one 1 in its first k positions and 0 for all other non-zero vectors.

Let us now try to count the number of codes with min. distance smaller than δn. At most, the number of such codes is the sum of $N(w)$ over all non-zero words w of weight less than or equal to δn.

$$\begin{aligned}
\#\{C \in S_0 | d(C) \leq \delta n\} &\leq \sum_{w:0<\|w\|\leq \delta n} N(w) \\
&= 2^{(k-1)(n-k)} \#\{w | 0 < \|w\| \leq \delta n \text{ and } w_{1,\ldots,k} \neq 0\} \\
&< 2^{(k-1)(n-k)} \#\{w | 0 < \|w\| \leq \delta n\} \\
&= 2^{(k-1)(n-k)} \sum_{i=1}^{\delta n} \binom{n}{i} \\
&\leq 2^{(k-1)(n-k)} 2^{nh(\delta)} \\
&= 2^{k(n-k)+k-n+nh(\delta)} \\
&= 2^{k(n-k)+n(R-C_\delta)}
\end{aligned}$$

In that there are a total of $2^{k(n-k)}$ possible matrices of size $k \times (n-k)$, the proportion of systematic codes of minimum distance smaller than δn is at most $2^{n(R-C_\delta)}$. This implies the theorem. QED.

Hence, we have the bound:

$$p_m < 2^{R-C_{\gamma\epsilon}}$$

2.2 Bob's test: Honest Alice

Bob test is based on the number of differences he expects to see between x, the word Alice claims she sent and x', the word he received. He accepts iff $\|x \oplus x'\| < (\epsilon + \lambda)n$.

Because x' is transmitted over a BS_ϵ, the weight of $x \oplus x'$ is the sum of n independent Bernouilli trials of probability ϵ. Hence, if Alice is honest, corollary 2 guarantees us that

$$p_h \leq e^{-n\frac{\lambda^2}{2\epsilon(1-\epsilon)(1+\lambda/2\epsilon)}}$$

as long as $\lambda < \epsilon$.

2.3 Bob's test: Cheating Alice

What if Alice tries to change her mind, i.e. tries to pass some fraudulent word off as the word she really sent? What are her chances of passing Bob's test with two different codewords?

Let x be the word Alice originally sent (not necessarily a codeword). We want to bound Alice's chances of being able to pass off two different codewords. Let y be the farther (from x) of the two codewords Alice would like to pass off as her own.

Let $z = \|y \oplus x\|$. Consider the random variables Y_i (for $i = 1, \ldots, n$) and D_y defined as follows:

$$\begin{aligned} Y_i &= \begin{cases} 1 & \text{if } x' \text{ and } y \text{ differ in position } i \\ 0 & \text{otherwise} \end{cases} \\ D_y &= \|y \oplus x'\| = \sum_i Y_i \end{aligned}$$

We can partition the Y_i's into to two sets, S_1 and S_2 where S_1 contains the Y_i's for which x and y differ in position i ($\#S_1 = z$) and S_2 contains the $n - z$ others. S_1 contains variables with expected value $1 - \epsilon$, while S_2 contains variables with expected value ϵ. Thus,

$$E(D_y) = \epsilon(n - z) + z(1 - \epsilon) = \epsilon n + z(1 - 2\epsilon).$$

We assume here that the code Bob chose was good, i.e. $d \geq \gamma\epsilon$. No matter what word x Alice really sends, she cannot find two codewords both of which are at distance less than $\gamma\epsilon/2$ from x. In that we chose y to be the farther of the two codewords Alice would like to be able to unveil, we know that $z \geq \gamma\epsilon/2$. Hence,

$$E(D_y) \geq \epsilon n + \gamma\epsilon(1/2 - \epsilon)$$

Because all of the Y_i's are Bernouilli variables with probability ϵ or $1 - \epsilon$, we can apply corollary 2:

$$Pr\left[D_y - E(D_y) \leq -\lambda_0 n\right] \leq e^{-n\frac{\lambda_0^2}{2\epsilon(1-\epsilon)(1+\lambda_0/2\epsilon)}}$$

In our situation, $\lambda_0 = E(D_y) - \epsilon n + \lambda$, where λ is from Bob's test. Because we cannot be guaranteed that $E(D_y)$ will be significantly larger than $\epsilon n + \gamma\epsilon(1/2 - \epsilon)$, we may as well assume that it isn't. Hence,

$$p_a \leq e^{-n\frac{\lambda_0^2}{2\epsilon(1-\epsilon)(1+\lambda_0/2\epsilon)}}$$

where $\lambda_0 = \gamma\epsilon(1/2 - \epsilon) - \lambda$.

2.4 Optimal value for λ

Clearly, as long as $0 < \lambda < \gamma\epsilon(1/2 - \epsilon)$, both p_a and p_h will be exponentially small in n. The question is, for fixed γ and ϵ, what is the best value for λ?

What we are trying to maximize is the probability that an honest Alice will pass the test *and* that a cheating Alice will be unable to unveil x as two different codewords. Hence, we want λ such that $p_{h,a} = (1 - p_h)(1 - p_a)$ is minimal.

We notice that $p_h(\lambda)$ and $p_a(\lambda_0)$ are identical functions which are strictly decreasing when their variable is positive. It is clear that when $\lambda_0 = \gamma\epsilon(1/2 - \epsilon) - \lambda$, the function $p_{h,a}(\lambda)$ will have a critical point at $\lambda = \gamma\epsilon(1/2 - \epsilon)/2$. We want to show that this is the absolute maximum of the function (in any case, it will be pretty damn close).

2.5 Bob's information

This discussion is based on Theorem 3 in [BBCM95]:

Theorem 5 (Privacy Amplification) *Let W be a random variable over some alphabet $\mathcal{W}$ with Renyi entropy $R(W)$. Let G be a random variable distributed uniformly over a universal$_2$ class of hash functions $\mathcal{W} \to \{0,1\}^r$. Then*

$$\begin{aligned} H(G(W)|G) \geq R(G(W)|G) &\geq r - \log_2(1 + 2^{r-R(W)}) \\ &\geq r - \frac{2^{r-R(W)}}{\ln 2} \end{aligned}$$

If X is the random variable representing the string Alice sends Bob through the BSC, we will call $BS_\epsilon(X)$ the r.v representing the string Bob receives and D_X the r.v. representing the Hamming distance between X and $BS_\epsilon(X)$. We will assume that what information Bob could get is actually of some use to him, that is we assume Alice sent a valid codeword in the first place. In our situation, the W of the previous theorem will be the r.v. $X|BS_\epsilon = v, D_X = d, syn(X) = 0$.

Given $v, w \in F_2^n$, consider the number of words of the same syndrome as w which are at the same distance as w from v. Because of the additivity of the syndrome, this is the same as the number of words at distance $\|v \oplus w\|$ from 0 which have syndrome $syn(w) \oplus syn(v)$. So define $N_d(S), S \in F_2^{n-k}$ to be the number of words of weight d which have syndrome S.

Assume Bob has an oracle which tells him the value of $D_X = d$. He also has $BS_\epsilon(X) = v$. How many possibilities does he have for the value of X? There are exactly $N_d(syn(v))$ different words of syndrome 0 at distance d from v (because $0 \oplus syn(v) = syn(v)$). The Renyi entropy of X from his point of view is thus:

$$R(X|BS_\epsilon(X) = v, D_X = d, syn(X) = 0) = \log_2 N_d(syn(v))$$

Hence, by 5, we can conclude that, for G randomly distributed over a *universal*$_2$ class of hash functions:

$$\begin{aligned} H(G(X)|G, BS_\epsilon(X) = v, D_X = d, syn(X) = 0) &\geq r - \frac{2^{r - \log_2 N_d(syn(v))}}{\ln 2} \\ &= r - \frac{2^r}{N_d(syn(v)) \ln 2} \end{aligned}$$

From this, we can calculate:

$$
\begin{aligned}
H(G(X)|G,BS_\epsilon(X),D_X=d,syn(X)=0) &= \frac{1}{\binom{n}{d}}\sum_{v:\|v\|=d} H(G(X)|G,BS_\epsilon(X)=v,D_X=d,syn(X)=0) \\
&\geq \frac{1}{\binom{n}{d}}\sum_{v:\|v\|=d} r-\frac{2^r}{N_d(syn(v))\ln 2} \\
&= \frac{1}{\binom{n}{d}}\sum_{Y\in\{0,1\}^{n-k}} N_d(Y)\left(r-\frac{2^r}{N_d(Y)\ln 2}\right) \\
&= r-\frac{1}{\binom{n}{d}}\sum_{Y\in\{0,1\}^{n-k}} N_d(Y)\frac{2^r}{N_d(Y)\ln 2} \\
&= r-\frac{2^{r+n-k}}{\binom{n}{d}\ln 2}
\end{aligned}
$$

By 2, we know $(\epsilon-\beta)n \leq D_X \leq (\epsilon+\beta)n$ with probability at least $1-2e^{-n\beta^2}$. By 3, this implies $\binom{n}{D_X} \geq \frac{2^{nh(\epsilon-\beta)}}{\sqrt{2n}}$. Averaging over D_X, we get:

$$
\begin{aligned}
H(G(X)|G,BS_\epsilon(X),D_X,syn(X)=0) &\geq \left(1-2e^{-n\beta^2}\right)\left(r-\frac{\sqrt{2n}2^{r+n-k}}{2^{nh(\epsilon-\beta)}\ln 2}\right) \\
&= \left(1-2e^{-n\beta^2}\right)\left(r-\frac{\sqrt{2n}2^{r+n[1-R-h(\epsilon-\beta)]}}{\ln 2}\right) \\
&= \left(1-2e^{-n\beta^2}\right)\left(r-\frac{\sqrt{2n}2^{r-n[R-C_{\epsilon-\beta}]}}{\ln 2}\right)
\end{aligned}
$$

This means that Bob's real average uncertainty (without the oracle) is even higher than this. Hence, α, Bob's average information, is bounded above.

We notice that the β used here is a pure mathematical construct used to get around possible extreme values of D_X. We are free to choose its value however we wish.

2.6 Picking the hash vector M

Most values of m will give good hash functions i.e. exactly one half of the codewords will hash to 1 and the other half to 0. However, a bad case can arrive (with probability exponentially small in n): m could be a dual codeword. This case could arrise but can be avoided by simply checking if $mG^T = 0$. This is supported by the following little lemma.

Lemma 6 (Good m are very likely) *The function f_m from a linear code C onto $\{0,1\}$ defined by $f_m(x) = x \odot m$ is balanced over its image. This image is $\{0\}$ if m is in the dual code D and $\{0,1\}$ otherwise.*

This is true because f_m is always a homomorphism of vector spaces (and hence balanced) and because if $m \notin$ the dual code, then there is at least one word x in the code such that $x \odot m = 1$.

2.7 Summary of results

Normally, ϵ is fixed, we choose: γ, δ, λ and β, where $R = k/n = C_\epsilon + \delta$ and:

$$
\begin{array}{rcl}
0 < & \epsilon & < 1/2 \\
0 < & \gamma & < 1 \\
0 < & \delta & < h(\epsilon) - h(\epsilon\gamma) \\
0 < & \lambda & < \gamma\epsilon(1/2 - \epsilon) \\
0 < & \beta & < \epsilon
\end{array}
$$

We get the following bounds:

$$
\begin{aligned}
p_h &\leq e^{-n\frac{\lambda^2}{2\epsilon(1-\epsilon)(1+\lambda/2\epsilon)}} && (1)\\
p_a &\leq e^{-n\frac{\lambda_0^2}{2\epsilon(1-\epsilon)(1+\lambda_0/2\epsilon)}} \text{ where } \lambda_0 = \gamma\epsilon(1/2-\epsilon) - \lambda && (2)\\
p_m &\leq 2^{n(h(\epsilon\gamma)-h(\epsilon)+\delta)} && (3)\\
\alpha &\leq 2e^{-n\beta^2} + \left(1 - 2e^{-n\beta^2}\right)\frac{\sqrt{2n}2^{1-n[h(\epsilon-\beta)-h(\epsilon)+\delta]}}{\ln 2} && (4)
\end{aligned}
$$

3 Questions

1. Can we refine the argument to get rid of the multiplicative factor $1 - 2e^{-n\beta^2}$?
2. What is the optimal value of β as a function of the other parameters?
3. Are there values of λ, γ and δ which are optimal? What are they?
4. How stable is the protocol when the value of ϵ is not known exactly? Just how much uncertainty still permits exponential bounds on the error probabilities? Do the optimal values of the parameters change in such a case?