Cryptographic Test Correction

Eric Levieil and **David Naccache**
École normale supérieure
Département d'informatique, Équipe de cryptographie
45 rue d'Ulm, F-75230, Paris CEDEX 05, France
{eric.levieil,david.naccache}@ens.fr

Abstract. Multiple choice questionnaires (MCQs) are a widely-used assessment procedure where examinees are asked to select one or more choices from a list.

This invited talk[1] explores the possibility of transferring a part of the MCQ's correction burden to the *examinee* when sophisticated technological means (*e.g.* optical character recognition systems) are unavailable. Evidently, such schemes must make cheating difficult or at least conspicuous.

We did not manage to devise a fully satisfactory solution (cheating strategies do exist) – but our experiments with a first clumsy system encouraged us to develop alternative MCQ formats and analyze their performance and security.

1 Foreword

Three years ago I moved from industry to academia.

At the first staff meeting, I discovered that the university's policy[2] *was to assign first-year amphitheater courses to the newest staff members. I was delighted by the perspective of lecturing computer science to 600 students.*

A day later, I got a call from the Reprography Department. The reprographer wanted to ascertain that the test's camera-ready copy will reach him at least a month before the test. I suddenly realized that my Ph.D. students and I will have to spend our winter vacations correcting a heap of 600 multiple choice questionnaires (MCQs).

While designing the MCQ, an intriguing question started taunting my mind: Could the freshmen "chip-farm" help correcting the heap of copies?

After all – since twenty years we routinely witness all sorts of miracles in cryptography: Alice and Bob regularly prove knowledge without revealing secrets, anonymously say "no", flip coins over the phone, transfer bits obliviously and so on.

Could any of these wonderful tools help?

I challenged my Ph.D. students to imagine methods for safely delegating to the examinees the burden of MCQ correction.

The result is the cryptographic curiosity presented here.

David Naccache

2 Introduction

MCQs are an assessment procedure, invented in 1914 by Frederick J. Kelly, where examinees are asked to select one or more choices from a list. MCQs are widely used in education, opinion polls, elections, and many other areas.

[1] This *is not a refereed research paper.*

[2] Université Paris II Panthéon-Assas

This paper explores the possibility of safely transferring a part of the MCQ's correction burden to the *examinee*, when sophisticated technological means, such as optical character recognition (OCR) systems, are unavailable.

We regard an MCQ as a list of n questions $\{\texttt{question}_1, \ldots, \texttt{question}_n\}$.

Each $\texttt{question}_i$ is associated to two potential choices $\texttt{answer}_{i,0}$ and $\texttt{answer}_{i,1}$, of which only one is correct. We denote by c the MCQ's answer-vector, namely:

$$c_i = 1 \text{ iff } \texttt{answer}_{i,1} \text{ is correct.}$$

The student is required to generate an answer-vector $\tilde{c}$:

$$\tilde{c}_i = 1 \text{ iff the student thinks that } \texttt{answer}_{i,1} \text{ is correct.}$$

And the corrector, usually the newest member of the faculty staff, computes the mark:

$$m = n - \sum_{i=1}^{n} (c_i \oplus \tilde{c}_i)$$

2.1 Cryptographic Test Correction

To transfer the correction burden to the examinee, the MCQ designer generates a secret key k and computes, using an *encoding algorithm* $\mathcal{E}$, a set of $2n$ public values $v_{i,j}$ where $1 \leq i \leq n$, $j \in \{0, 1\}$:

$$\{v_{i,j}\} = \mathcal{E}(c, k)$$

Students are instructed to:

- Generate $\tilde{c}$ as before but, in addition, apply an easily computable *accumulation algorithm* $\mathcal{M}$ to $\{v_{i,j}\}$ and $\tilde{c}$.
- Write down the result $t = \mathcal{M}(\{v_{i,j}\}, \tilde{c})$ on the questionnaire.

The examiner uses a (potentially complex) *scoring algorithm* $\mathcal{C}$ to compute the student's final mark m:

$$m = \mathcal{C}(t, k) = \begin{cases} n - \sum_{i=1}^{n} (c_i \oplus \tilde{c}_i) & \text{if } \exists \tilde{c} \text{ such that } t = \mathcal{M}(\{v_{i,j}\}, \tilde{c}) \\ \perp & \text{otherwise} \end{cases}$$

We call $\{\mathcal{E}, \mathcal{M}, \mathcal{C}\}$ a *Cryptographic Test Correction* (CTC) scheme.

2.2 Desirable Features

Ideally, we would like $\{\mathcal{E}, \mathcal{M}, \mathcal{C}\}$ to have the following features:

Security: We say that an algorithm $\mathcal{A}$ has a CTC *cheating advantage* ϵ if:

$$\left| \Pr[\mathcal{C}(\mathcal{A}(\{v_{i,j}\}, \tilde{c}), k) > n - \sum_{i=1}^{n} c_i \oplus \tilde{c}_i] - \frac{1}{2} \right| \geq \epsilon$$

$\{\mathcal{E}, \mathcal{M}, \mathcal{C}\}$ is $\{w, \epsilon\}$*-secure* if no algorithm requiring w basic calculator operations (*i.e.* $+, -, \times, \div$) has a CTC cheating advantage ϵ.

In other words, we require that even if a cheating student knows the correct answers to all the questions but one, inferring the missing answer from $\{v_{i,j}\}$, or (more generally) manipulating t to artificially increase m is unfeasible given the simple calculator authorized by the university's regulations (Figure 1) and the test's limited duration.

Unlike e-cash or e-voting protocols, CTC does not seem to require protection against colluding parties (examinees cannot communicate). However, we do need some form of limited resistance against adaptive attacks as students knowing u correct answers can potentially generate 2^u valid t-values corresponding to marks expectedly[3] ranging between zero and $\frac{(n+u)}{2}$.

Efficiency: Trivially, one can design a secure CTC by assigning to the $v_{i,j}$ successive powers of two or zeros. *i.e.*:

$$v_{i,j} = \begin{cases} 0 & \text{if } j = 0 \\ 2^{i-1} & \text{if } j = 1 \end{cases}$$

The encoding $v_{i,j} = j \times 2^i$ is secure but inefficient. The size of t, *i.e.* n bits, is obviously an *overkill* as we do not need to convey to the examiner the *precise* answer vector $\tilde{c}$ but only the Hamming distance between c and $\tilde{c}$ (a quantity of information encodable in $\log_2 n$ bits).

Denoting by T the maximal bit length of t we require that $T < n$.

T measures the CTC's efficiency as it represents the number of digits that the corrector will need to key into his computer per corrected form.

As the theoretical foundations were ready, we started thinking about implementing CTCs.

3 Practical Experiments with an Insecure and Clumsy CTC

A simplified CTC was tested on 550 economics freshmen[4]. To avoid unresolvable complaints and computational errors, students were requested to both tick the correct answers and use the CTC. Ticked answers were used whenever $\mathcal{C}$ returned $\perp$ (27 cases), when a statistical alert occurred (unrecorded number of cases) or when the student didn't sum up the $v_{i,j}$ at all (79 cases).

We made the following *risk management* assumptions:

- As modular arithmetic was not part of the students' curriculum we assumed that the theoretical tools necessary for cheating were not at the average student's command.
- No parameters or specifications were revealed and a form of psychological warfare was used: we subtly hinted that the scheme is "...probably very resilient to cheating...".
- A cheater who would have discovered[5] one of the (many) existing cheating strategies would have anyway obtained an excellent mark given the course's subject matter[6].

3.1 Description

Generate five integers $\{\rho, k, g > nk, p > (n+1)g, e\}$ such that $\gcd(e, p) = 1$.

The authorized pocket-calculator must be able to handle at least the number $(\rho + 1)np$.

Prepare the following values:

- Pick n random bits $\{b_1, \ldots, b_n\}$ and define $\epsilon_{i,b_i} = 0$ and $\epsilon_{i,1-b_i} = 1$.

[3] The student can *force* part of the MCQ to contribute any precise number of points $\leq u$. Answers to the rest of the MCQ will result in an expected contribution of $\frac{(n-u)}{2}$ points.
[4] Examinees were given additional thirty minutes to account for the extra computational burden.
[5] *e.g.* given the scheme's additive nature.
[6] *Introduction to Computer Science*

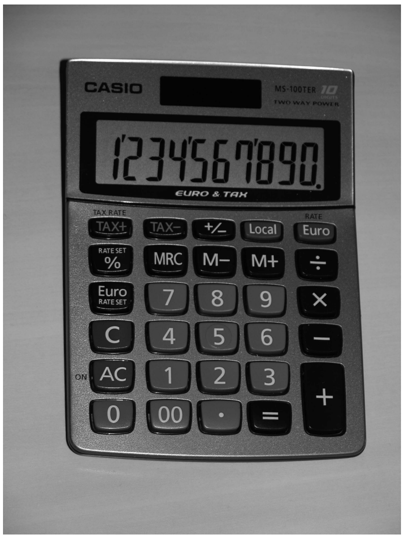

Fig. 1. Authorized Calculator (10-Digit Precision, Restricted to $+, -, \times, \div$).

- For $1 \leq i \leq n$ and $j \in \{0, 1\}$ generate randomly $0 \leq r_{i,j} \leq \rho$.
- For $1 \leq i \leq n$ generate randomly $0 \leq a_i < p$.

We denote by $\tau_i = (\neg c_i \oplus \tilde{c}_i)k$, in other words:

$$\tau_i = \begin{cases} k & \text{if the student's answer to question } i \text{ is correct} \\ 0 & \text{if the student's answer to question } i \text{ is incorrect} \end{cases}$$

and define:

$$v_{i,j} = ((a_i + (\neg c_i \oplus j)k + g\epsilon_{i,j})\, e \mod p) + r_{i,j} \times p$$

Students were instructed to sum the $v_{i,j}$ corresponding to their answers and *answer randomly* whenever they don't know the answer[7].

The examiner computes: $\left(t \times e^{-1} - (\sum_{i=1}^{n} a_i) \mod p\right)$ which is $\sum_{i=1}^{n}(\tau_i + g\epsilon_{i,\tilde{c}_i}) \in \mathbb{N}$.

This is easily checked by bounding:

$$0 < \sum_{i=1}^{n}(\tau_i + g\epsilon_{i,\tilde{c}_i}) < n(k + g) = g + n \times g < p$$

We therefore recover the exact value:

$$t' = t \times e^{-1} - \left(\sum_{i=1}^{n} a_i\right) \mod p = \sum_{i=1}^{n}(\tau_i, + g\epsilon_{i,\tilde{c}_i}) = mk + g\sum_{i=1}^{n}\epsilon_{i,\tilde{c}_i} = mk + gq$$

where:

$$0 \leq q = \sum_{i=1}^{n}\epsilon_{i,\tilde{c}_i} \leq n$$

[7] the rationale is both the need to collect all the a_is for decryption to work, and preventing "the cryptanalyst" from generating t-values corresponding to *precisely* chosen marks.

but $mk \leq nk < g$ hence we can retrieve mk and q with no ambiguity.

$$q = \left\lfloor \frac{t'}{g} \right\rfloor \quad \text{and} \quad m = \frac{t' - qg}{k}$$

If $m \notin \mathbb{N}$ or $m \notin [0, n]$ or $q \notin [0, n]$ return $\perp$ (*i.e.* trigger a manual form verification). The odds to hit a multiple of k by picking t at random are $\frac{1}{k}$.

Implementation values and a marking example are given in Appendix A.

Fig. 2. 550 Distrusted Correctors (Right) Filling 550 Cryptographic MCQs (Left).

Fig. 3. The University's Grand Amphithéatre.

3.2 Statistical Analysis

Unfortunately, this scheme is insecure. Namely, if a student knows the algorithm's specifications, then several efficient cheating strategies exist. For instance the cheater may identify one correct answer, say i, subtract the incorrect $v_{i,j}$ from the correct one and obtain a "clean" encoding of $+k$:

$$\Delta = (k + \epsilon g)e + \alpha p \quad \text{where} \quad \epsilon \in \{-1, 1\}$$

The cheater will then pick random answers to the entire questionnaire, thereby reaching an expected average mark of $\frac{n}{2}$ and artificially improve it by adding a multiple of Δ.

To overcome this (to some extent) we used a basic statistical test on q. Namely, if q does not exceed a given likelihood threshold, we treat the form as suspicious and verify it manually. Indeed, if the cheater brutally adds $\mu\Delta$ to t the additional $\pm\mu g$ will start showing up as a statistical bias in the distribution of q.

Evidently, a very good student could use much smarter cheating strategies based on the linear combination of several Δ values derived from different questions weighted by moderate coefficients but we considered such a strategy unlikely given our risk management assumptions.

A given $v_{i,j}$ has a $\frac{1}{2}$ probability to contain no g and a $\frac{1}{2}$ probability to contain g. Thus, the probability that q takes a given value $0 \le d \le n$ is simply:

$$\Pr[q = d] = \binom{n}{d} \times \frac{1}{2^n}$$

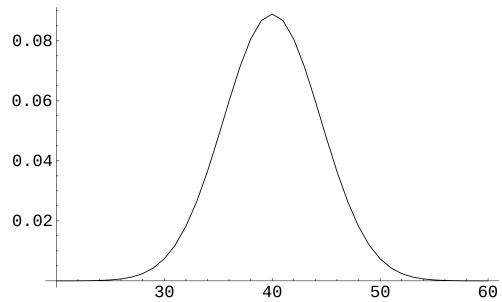

Fig. 4. $\Pr[q = d] = \binom{80}{d} \times 2^{-80}$

That is, for $n = 80$:

d	$\Pr[\,\lvert q - n/2\rvert \le d\,]$	d	$\Pr[\,\lvert q - n/2\rvert \le d\,]$	d	$\Pr[\,\lvert q - n/2\rvert \le d\,]$
0	0.08893	7	0.90709	14	0.99895
1	0.26245	8	0.94334	15	0.99955
2	0.42357	9	0.96701	16	0.99982
3	0.56596	10	0.98168	17	0.99993
4	0.68569	11	0.99032	18	0.99997
5	0.78148	12	0.99513	19	0.99999
6	0.85436	13	0.99768	20	1.00000

Table 1. $\Pr[q = d] = \binom{80}{d} \times 2^{-80}$.

We hence triggered, in addition, a manual verification whenever $|q - 40| \ge 7$.

We conjecture that no student tried to cheat but the scheme's clumsiness and poor security performances motivated the quest for alternative CTC mechanisms – some of which we describe in the next section.

4 Alternative CTC Mechanisms

An alternative line of research is the development of new MCQ mechanisms. This section describes such a scheme – called *Interval Estimation* MCQs (IEMCQs).

Again, $\texttt{question}_i$ is associated to two potential choices $\texttt{answer}_{i,0}$ and $\texttt{answer}_{i,1}$, of which only one is correct. $\texttt{answer}_{i,0}$ is printed in *blue* while $\texttt{answer}_{i,1}$ is printed in *red*[8].

The test's idea consists in having the student determine the (correct) number of (correct) red answers.

In other words, the student's output is a sequence of three digits: the number of red answers, the number of blue answers and (implicitly) the difference between n and the sum of the previous two, *i.e.* the number of unsolved questions. This output can be encoded using only two integers – we choose to ask for an interval containing the number of red answers.

Assume, for example, that $n = 9$ and that the examinee identified 2 reds and 3 blues, the student's answer will be $[2, 6]$. This notation means that the student thinks that there are at least 2 reds and at most $6 = 9 - 3$ reds. The low and high bounds will be denoted by a and c (here $a = 2$ and $c = 6$) while b will denote the correct answer, *i.e.* the precise number of reds. In other words, $[a, c]$ reads as *"I hope that $a \leq b \leq c$"*. The interval's narrowness reflects the examinee's knowledge.

Evidently, if questions are independent, we would expect $b \simeq \frac{n}{2}$. Hence, we must first pick b randomly in $[0, n]$ and color the IEMCQ accordingly. In practice, we recommend $n = 9$, as this shrinks answers to two decimal digits (compact notation) and allows approaching 100 points using eleven question-packs. Note that, unlike additive CTCs, filling an IEMCQ does not require a pocket calculator.

Mapping $[a, c]$ to a mark (scoring) is the most delicate part, as the scoring function must:

- faithfully reflect the student's knowledge.
- be fairly resilient to statistical attacks.
- and have a small standard deviation.

In addition – we would like IEMCQs to allow students who know answers with sufficiently high probability (say 80%) to continue benefiting from this knowledge.

As these objectives are independent and incomparable, an "ideal" scoring function might not exist. We hence looked for functions that *reasonably comply* with the above objectives. The following proposals are thus examples and not reference designs.

We will start with a basic scoring function $\mathcal{C}_1$ and refine it progressively, explaining at each step the rationale of our successive refinements. To simplify calculations we assume that a correct answer is rewarded by a point while an incorrect answer is penalized by a point.

4.1 Notations and definitions

We denote by $\chi_{a,c}(x)$ the Heaviside function:

$$\chi_{a,c}(x) = \begin{cases} 1 & \text{if } x \in [a, c] \\ 0 & \text{otherwise} \end{cases}$$

and by $d_{a,c}(x)$ the distance between x and the interval $[a, c]$, *i.e.*:

$$d_{a,c}(x) = (1 - \chi_{a,c}(x)) \max(a - x, x - c)$$

[8] The use of colors is not mandatory. Any form of distinction between answers will do (*e.g.* preceding answers by symbols such as ♡ or ♠ *etc.*).

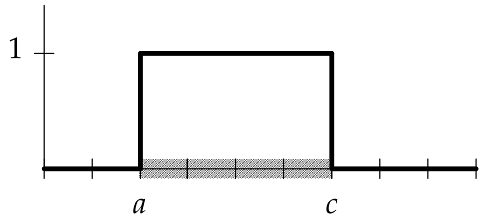

Fig. 5–A. The Heaviside Function $\chi_{a,c}(x)$.

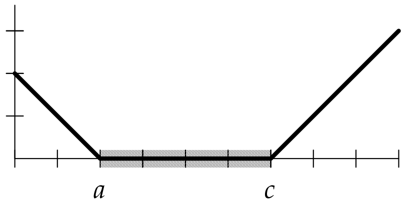

Fig. 5–B. The Distance Function $d_{a,c}(x)$.

We also define two auxiliary variables:

$$\Delta = n + a - c \quad \text{and} \quad \delta = \begin{cases} \left|\frac{a}{\Delta} - \frac{b}{n}\right| & \text{if } \Delta \neq 0 \\ 0 & \text{if } \Delta = 0 \end{cases}$$

Δ is the number of possibilities that the student has ruled out.

δ expresses the difference between the ratio of reds estimated by the student ($\frac{a}{\Delta}$) and the *actual* ratio of reds ($\frac{b}{n}$) in the IEMCQ.

4.2 Heaviside Scoring

Heaviside scoring is defined as:

$$\mathcal{C}_1(n, a, b, c) = \Delta + (\chi_{a,c}(b) - 1)(n + 1)$$

Intuitively, $\mathcal{C}_1$ correlates the student's mark to the number of possibilities ruled-out. The role of the *penalty component* $(\chi_{a,c}(b) - 1)(n + 1)$ is to equate the expectation of random guessing to zero.

$\mathcal{C}_1$ complies with all criteria but resilience to statistical attacks. Indeed, a cheater could use the proportion of reds he spots as an estimate (sample) of the actual ratio of reds in the IEMCQ (IEMCQ "redness") and narrow his interval accordingly. This might significantly optimize his mark (*e.g.* by +20%).

For example, if the cheater successfully detected 3 reds and no blues amongst $n = 9$, the risk taken by betting that the unknown answers contain 2 more reds is moderate. We call such cheaters "narrowers".

4.3 Distance Scoring

In addition, $\mathcal{C}_1$'s penalty component is insensitive to the *magnitude of mistakes*. After all, it would be desirable to penalize a $\{[a, c] = [1, 4], b = 5\}$ less than a $\{[a, c] = [1, 4], b = 9\}$.

While it seems clear that gradual penalty implies using $d_{a,c}(x)$, there seems to be no obvious way to tune the penalty function (other than increasing penalty as $d_{a,c}(x)$ grows). We therefore used the probability $\varphi(d)$ to miss b by d to fine-tune a linear penalty coefficient γ_1:

$$\mathcal{C}_2(n, a, b, c) = \Delta - \gamma_1 \ (n + 1) \ d_{a,c}(b)$$

Note that $\varphi(x)$ reflects the test's hardness (*i.e.* depending on *pedagogic* factors).

Typically, the configurations $\varphi(1) = \varphi(2) = \frac{1}{2}$ or $\{\varphi(1) = \frac{6}{10}, \varphi(2) = \frac{3}{10}, \varphi(3) = \frac{1}{10}\}$ are $\mathcal{C}_1$-compatible when $\gamma_1 = \frac{2}{3}$. We recommend to adopt this value of γ_1 – a value we used in our simulations hereafter.

A second design objective is to discourage narrowers. Indeed, an examinee's answer is not only an interval. It also expresses a redness approximation.

In general a (non exaggerating) narrower will score the same $\varDelta$ as an honest examinee, however, the narrower's redness estimate will be less accurate. In other words, his δ will be *expectedly bigger*. We thus use δ to damp $\varDelta$:

$$\mathcal{C}_3(n,a,b,c) = \varDelta(1-\delta) - \gamma_1\ (n+1)\ d_{a,c}(b)$$

4.4 Father Christmas Scoring

During the French revolution, different strategies for abolishing birth privileges were debated. Proposals ranged from forbidding titles to exiling noblemen or... making titles available to anybody *i.e.* eliminate distinctions by devaluation.

All our scoring functions allow cheaters to estimate the IEMCQ's redness. While endeavoring to limit the cheaters' redness estimation abilities (using δ) we also reduce the cheaters' advantage by devaluation: namely, we award automatically to any examinee the cheaters' redness approximation advantage. We call this "*Father Christmas Scoring*", as we distribute extra points to all examinees.

$$\mathcal{C}_4(n,a,b,c) = \begin{cases} \mathcal{C}_3(n,a,b,c) + \gamma_2(c-a) & \text{if } b=c=n \text{ or } a=b=0 \\ \mathcal{C}_3(n,a,b,c) & \text{otherwise} \end{cases}$$

$\mathcal{C}_4$'s side-effect is an increase in standard deviation, but this increase can be controlled by γ_2. We propose to use $\gamma_2 = \frac{1}{2}$.

4.5 Features

Accuracy Table 2 shows the correlation between the mark obtained by considering a test as a traditional MCQ and as an IEMCQ scored with $\mathcal{C}_\ell$ (for $\ell = 1,3,4$).

The quantity:

$$\mu_{k,n} = \sum_{a=0}^{k}\sum_{b=0}^{n}\binom{b}{a}\binom{n-b}{k-a} = (k+1)\binom{n+1}{k+1}$$

counts the number of different ways in which k correct answers can be potentially distributed between a reds and $k-a$ blues[9]. We can hence compute $\texttt{Av}[\mathcal{C}_\ell,k,n]$, the average mark of an examinee knowing k answers out of n in an IEMCQ scored with $\mathcal{C}_\ell$:

$$\texttt{Av}[\mathcal{C}_\ell,k,n] = \frac{1}{n\times\mu_{k,n}}\sum_{a=0}^{k}\sum_{b=0}^{n}\binom{b}{a}\binom{n-b}{k-a}\mathcal{C}_\ell(n,a,b,n-k+a)$$

Note that for $\mathcal{C}_1$ averaging is unnecessary as $\mathcal{C}_1$ coincides with scores obtained using a traditional MCQ.

[9] μ_k is the denominator of the k-th element in line n in Leibniz's Harmonic triangle

k	$\texttt{Av}[\mathcal{C}_1, k, 9]$	$\texttt{Av}[\mathcal{C}_3, k, 9]$	$\texttt{Av}[\mathcal{C}_4, k, 9]$
0	0.000	0.000	0.100
1	0.111	0.078	0.167
2	0.222	0.180	0.257
3	0.333	0.286	0.353
4	0.444	0.394	0.450
5	0.556	0.505	0.550
6	0.667	0.620	0.653
7	0.778	0.735	0.757
8	0.889	0.856	0.867
9	1.000	1.000	1.000

k	$\texttt{Av}[\mathcal{C}_1, k, 12]$	$\texttt{Av}[\mathcal{C}_3, k, 12]$	$\texttt{Av}[\mathcal{C}_4, k, 12]$
0	0.000	0.000	0.077
1	0.083	0.058	0.128
2	0.167	0.133	0.197
3	0.250	0.212	0.269
4	0.333	0.292	0.343
5	0.417	0.373	0.418
6	0.500	0.457	0.496
7	0.583	0.540	0.572
8	0.667	0.626	0.651
9	0.750	0.712	0.731
10	0.833	0.800	0.813
11	0.917	0.891	0.898
12	1.000	1.000	1.000

Table 2. Average Accuracy for $n = 9$ and $n = 12$.

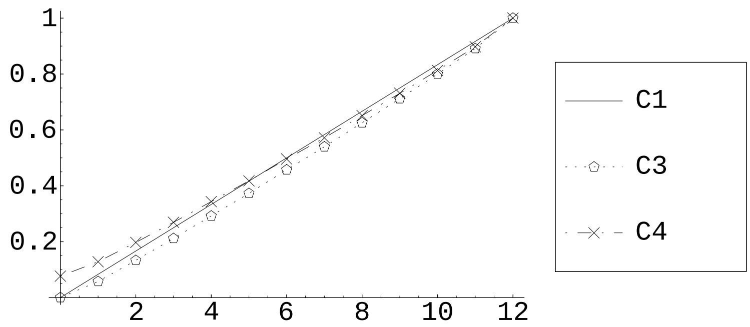

Fig. 5. $\texttt{Av}[\mathcal{C}_1, k, 12]$, $\texttt{Av}[\mathcal{C}_3, k, 12]$ and $\texttt{Av}[\mathcal{C}_4, k, 12]$

It appears that all scoring functions approximate quite faithfully a traditional MCQ (plain black line).

Narrowers' Advantage Table 3 lists $\texttt{Ad}[\mathcal{C}_\ell, k, n]$, the average advantage of a narrower over an honest examinee assuming that both know k answers (of which a are red).

The cheater's strategy will depend on $\{a, k\}$ – whose values he knows. As b is unknown to the cheater, we exhaust all the possible fraudulent answers $[\tilde{a}, \tilde{c}]$ (given $\{a, k\}$), select the best-performing (over $[\tilde{a}, \tilde{c}]$) cheating advantage:

$$\mathcal{F}_\ell(n, \tilde{a}, \tilde{c}, a, b, k) = \mathcal{C}_\ell(n, \tilde{a}, b, \tilde{c}) - \mathcal{C}_\ell(n, a, b, n - k + a)$$

and average[10] over b:

$$\texttt{Ad}[\mathcal{C}_\ell, k, n] = \frac{1}{n \times \mu_{k,n}} \sum_{a=0}^{k} \left(\max_{\substack{0 \leq \tilde{a} \leq n \\ \tilde{a} \leq \tilde{c} \leq n}} \left(\sum_{b=0}^{n} \binom{b}{a} \binom{n-b}{k-a} \mathcal{F}_\ell(n, \tilde{a}, \tilde{c}, a, b, k) \right) \right)$$

[10] The $\sum_{b=0}^{n}$ in the following formula can be simplified into a $\sum_{b=a}^{n-k+a}$.

k	0	1	2	3	4	5	6	7	8	9
$\mathtt{Ad}[\mathcal{C}_1, k, 9]$	0.000	0.198	0.198	0.175	0.147	0.102	0.069	0.031	0.000	0.000
$\mathtt{Ad}[\mathcal{C}_3, k, 9]$	0.012	0.091	0.145	0.144	0.134	0.102	0.074	0.038	0.008	0.000
$\mathtt{Ad}[\mathcal{C}_4, k, 9]$	0.000	0.068	0.111	0.110	0.101	0.078	0.052	0.027	0.000	0.000

k	0	1	2	3	4	5	6	7	8	9	10	11	12
$\mathtt{Ad}[\mathcal{C}_1, k, 12]$	0.000	0.208	0.225	0.216	0.205	0.177	0.151	0.113	0.082	0.049	0.020	0.000	0.000
$\mathtt{Ad}[\mathcal{C}_3, k, 12]$	0.011	0.081	0.144	0.167	0.163	0.156	0.136	0.110	0.086	0.054	0.028	0.005	0.000
$\mathtt{Ad}[\mathcal{C}_4, k, 12]$	0.000	0.066	0.118	0.142	0.136	0.131	0.111	0.091	0.068	0.042	0.022	0.000	0.000

Table 3. Narrower's Advantage for $n = 9$ and $n = 12$.

Table 2 reads as follows: Under $\mathcal{C}_1$ and $n = 9$, an honest examinee knowing $k = 2$ answers will score 0.22 (*cf.* to Table 1). Table 2 shows that under identical circumstances a cheater could hope to score $0.22 + 0.198 \simeq 0.42$.

Naturally, an ideal scoring function $\mathcal{C}_\ell$ will feature an $\mathtt{Ad}[\mathcal{C}_\ell, k, n] = 0$. Note that, for $n = 9$ and $n = 12$, we nearly always have:

$$\mathtt{Ad}[\mathcal{C}_4, k, n] \leq \mathtt{Ad}[\mathcal{C}_3, k, n] \leq \mathtt{Ad}[\mathcal{C}_1, k, n]$$

Partial Knowledge Another interesting benchmark is $\mathtt{Pa}[\mathcal{C}_\ell, \omega, n]$, the mark expected by an examinee who knows the answer to each question with probability ω.

We regard the experiment as a vision test where the student – standing at a distance from the corrector's answer form – tries to identify (and count) the colors of the IEMCQ's answers. As distance increases, ω tends to $\frac{1}{2}$, *i.e.* reds and blues become less and less distinguishable.

Having stared at the distant form for long enough, the student finally makes his mind and bets that the form contains s red answers and $n - s$ blue answers. The probability ω applies to each individual answer.

For each $\{\mathcal{C}_\ell, \omega, s, n\}$ there exists an optimal answer $[a, c]$ that we discover by exhausting all intervals $[\tilde{a}, \tilde{c}]$. The frequency-weighted score-contribution of these *optima* when the student's blind shot hits x reds amongst b reds and $s - x$ reds amongst $n - b$ blues gives:

$$\mathtt{Pa}[\mathcal{C}_\ell, \omega, n] = \frac{1}{n \times \nu_n} \sum_{s=0}^{n} \max_{\substack{0 \leq \tilde{a} \leq n \\ \tilde{a} \leq \tilde{c} \leq n}} \sum_{x=0}^{s} \sum_{b=0}^{n} \omega^{n-b-s+2x} (1-\omega)^{b+s-2x} \binom{b}{x} \binom{n-b}{s-x} \mathcal{C}_\ell(n, \tilde{a}, b, \tilde{c})$$

The normalization factor ν_n is:

$$\nu_n = \sum_{s=0}^{n} \sum_{x=0}^{s} \sum_{b=0}^{n} \omega^{n-b-s+2x} (1-\omega)^{b+s-2x} \binom{b}{x} \binom{n-b}{s-x}$$

ω	1.00	0.90	0.80	0.70	0.60	0.50
$\mathtt{Pa}[\mathcal{C}_1, \omega, 9]$	1.00	0.64	0.47	0.31	0.15	0.00
$\mathtt{Pa}[\mathcal{C}_3, \omega, 9]$	1.00	0.60	0.38	0.18	0.05	0.01
$\mathtt{Pa}[\mathcal{C}_4, \omega, 9]$	1.00	0.60	0.40	0.22	0.11	0.10
$\mathtt{Pa}[\text{MCQ}, \omega, 9]$	1.00	0.80	0.60	0.40	0.20	0.00

ω	1.00	0.90	0.80	0.70	0.60	0.50
$\mathtt{Pa}[\mathcal{C}_1, \omega, 12]$	1.00	0.67	0.50	0.33	0.17	0.00
$\mathtt{Pa}[\mathcal{C}_3, \omega, 12]$	1.00	0.61	0.40	0.20	0.05	0.01
$\mathtt{Pa}[\mathcal{C}_4, \omega, 12]$	1.00	0.62	0.42	0.22	0.10	0.08
$\mathtt{Pa}[\text{MCQ}, \omega, 12]$	1.00	0.80	0.60	0.40	0.20	0.00

Table 4. $\mathtt{Pa}[\mathcal{C}_\ell, \omega, 12]$ **for $n = 9$ and $n = 12$.**

Note that $\mathtt{Pa}[\mathcal{C}_\ell, \omega, n] = \mathtt{Pa}[\mathcal{C}_\ell, 1 - \omega, n]$ and $\mathtt{Pa}[\text{usual MCQ}, \omega, n] = \omega - (1 - \omega) = 2\omega - 1$.

Standard Deviation To assess the typical standard deviation of the different $\mathcal{C}_\ell$s the following simulation was performed: We generated one million random 99-question IEMCQs. Each IEMCQ contained 11 groups of $n = 9$ questions.

For each IEMCQ we generated a random binary vector $e_1, \ldots, e_{99}$. If $e_i = 1$ we considered that the examinee answered the i-th question correctly. If $e_i = 0$ the question was not answered. The IEMCQ was then corrected as a traditional MCQ and as an IEMCQ scored with $\mathcal{C}_1$, $\mathcal{C}_3$ and $\mathcal{C}_4$.

The experiment's means, μ and standard deviations, σ, are reported here:

	MCQ	$\mathcal{C}_1$	$\mathcal{C}_3$	$\mathcal{C}_4$
σ	0.050	0.050	0.052	0.060
μ	0.500	0.500	0.453	0.503

Table 5. Experimental Results.

Efficiency Table 5 allows to estimate efficiency, *i.e.* the number of decimal digits that the examiner needs to key into his computer per corrected form.

The examiner starts by setting a target σ' and multiplies the number of questions by:

$$\left(\frac{\sigma}{\sigma'}\right)^2$$

The following table assumes binary encoding for the traditional MCQ and the compressed answer encoding of Appendix B for $n = 12$:

	MCQ	$\mathcal{C}_1$	$\mathcal{C}_3$	$\mathcal{C}_4$
$n = 9$	31	24	24	32
$n = 12$	31	18	18	24

Table 6. Efficiency.

5 Further Research

It seems that homomorphism, necessary for mark accumulation, is the root-cause of the security problems encountered while designing all additive CTCs we could think of. The design of an additive CTC which is simultaneously practical, secure and efficient remains an open problem. Potential solutions could involve the use of non commutative operations such as moderate-size matrix multiplications or vector products[11]. Unfortunately, the cost of 80 matrix multiplications or vector products is prohibitive and so are the foreseeable error odds. The use of simple physical accessories (scratch cards [1], tables, envelopes, etc) also seems a promising idea.

The generalization of IEMCQs and scoring functions to more than two colors, attacks on the IEMCQs proposed in this paper or the development of better scoring functions are also welcome – as these might find practical applications during the 2008-2009 academic year...

6 Acknowledgments

The authors wish to warmly thank Nora Dabbous, Vanessa Gratzer, Hervé Leplat and Gueorgui Tzotchev for their comments and suggestions during the design of the schemes proposed in this work.

References

1. T. Moran and M. Naor, *Polling with Physical Envelopes: A Rigorous Analysis of a Human-Centric Protocol*, Advances in Cryptology - Eurocrypt 2006, Lecture Notes in Computer Science vol. 4004, pp. 88–108, Springer-Verlag.

[11] Taking advantage of the fact that $\vec{u} \wedge (\vec{v} \wedge \vec{w}) \neq (\vec{u} \wedge \vec{v}) \wedge \vec{w}$.

A Implementation Details

Fix $\{n = 80, g = 9189, k = 54, p = 3931231, e = 2032603\}$ and generate:

i	a_i	$v_{i,0}$	student		$v_{i,1}$
1	5498	50178050	•		18103810 ✓
2	19893	61139595	•		09409200 ✓
3	6294	✓ 32424036	•		04908839
4	6545	71173575		•	39099335 ✓
4	5441	✓ 32286548		•	67671047
5	9189	28139589		•	55033814 ✓
7	17580	✓ 68719202	•		81137287
8	13388	✓ 19850231	•		79443088
9	14708	✓ 61409445	•		49619172
10	19321	14960283		•	69373125 ✓
11	6861	44856367		•	72371564 ✓
12	1571	71821899		•	60024786 ✓
13	13903	✓ 05518892	•		09453543
14	18627	66751733		•	26815031 ✓
15	11471	23445338		•	62754228 ✓
16	14564	47835434	•		43900783 ✓
17	2659	42802779	•		61834542 ✓
18	11202	19495495	•		66045875 ✓
19	13374	70642801	•		34637330 ✓
20	10978	✓ 39557468		•	51354581
21	18810	61319906	•		21383204 ✓
22	13683	57926475	•		21921004 ✓
23	13811	78294568		•	26564173 ✓
24	12734	43495725		•	19283947 ✓
25	9648	60541981		•	01570096 ✓
26	12917	✓ 64958123		•	53788822
27	3219	72142831		•	09239715 ✓
28	8971	17157059		•	21084870 ✓
29	4619	✓ 67330650	•		67955042
30	1482	✓ 63890976		•	16719624
31	13212	✓ 24095841	•		35892954
32	11850	15728623		•	58347772 ✓
33	9833	31656743	•		31653323 ✓
34	5271	09108400	•		01242518 ✓
35	9059	✓ 54187901		•	19431214
36	10894	02794576	•		61138649 ✓
37	1410	07965293	•		39411721 ✓
38	6456	31796224		•	15446908 ✓
39	6519	06532204		•	49151353 ✓
40	5459	✓ 49217247	•		41358205

i	a_i	$v_{i,0}$	student		$v_{i,1}$
41	4395	✓ 36600526		•	49018611
42	2457	✓ 48613553	•		76135590
43	6430	37606525	•		80846646 ✓
44	18139	14405678	•		68818520 ✓
45	9341	61598589	•		81251324 ✓
46	3423	26839816	•		58286244 ✓
47	13508	75687895	•		78994734 ✓
48	4543	✓ 38652214	•		82520147
49	18648	15086852		•	49843539 ✓
50	10242	✓ 09910823		•	25639167
51	3981	✓ 32765573		•	72081303
52	4790	57477648	•		22093149 ✓
53	10402	68117501		•	43905723 ✓
54	13061	35916405	•		51016937 ✓
55	5825	22942575	•		65561724 ✓
56	1062	✓ 47239433	•		59657518
57	18333	11676329	•		81814095 ✓
58	19114	✓ 69576507	•		38130079
59	3226	63094152	•		42813605 ✓
60	15857	✓ 53546130	•		69895446
61	10718	73560627	•		69005004 ✓
62	7214	✓ 58360971	•		03948129
63	4281	13552933		•	17480744 ✓
64	18135	41656345		•	68550570 ✓
65	2170	27736431	•		27112039 ✓
66	4245	✓ 34725349	•		58316155
67	849	03800769		•	43109659 ✓
68	10077	32276769	•		12617194 ✓
69	927	✓ 24436812		•	25061204
70	7304	25391442	•		25388022 ✓
71	8668	73518851		•	34203121 ✓
72	18606	24067070		•	47030064 ✓
73	10119	82265016		•	78330365 ✓
74	7537	✓ 70480342	•		27240221
75	5030	42415286		•	49653356 ✓
76	18830	✓ 03377285		•	46624246
77	3049	✓ 76476460	•		48961263
78	17663	60833762		•	21518032 ✓
79	15458	✓ 40577426		•	17614432
80	6769	15416617	•		22654687 ✓

i	$\epsilon_{i,0}$	$r_{i,0}$	$r_{i,1}$
1	1	12	4
2	1	15	2
3	1	8	1
4	1	18	9
5	1	8	17
6	1	7	13
7	0	17	20
8	0	5	20
9	1	15	12
10	1	3	17
11	0	11	18
12	0	18	15
13	1	1	2
14	1	16	6
15	0	5	15
16	0	12	11
17	1	10	15
18	1	4	16
19	1	17	8
20	1	10	13

i	$\epsilon_{i,0}$	$r_{i,0}$	$r_{i,1}$
21	1	15	5
22	1	14	5
23	1	19	6
24	1	11	4
25	0	15	0
26	0	16	13
27	0	18	2
28	0	4	5
29	0	17	17
30	1	16	4
31	1	6	9
32	1	4	14
33	0	8	8
34	0	2	0
35	0	13	4
36	1	0	15
37	0	2	10
38	1	8	3
39	1	1	12
40	1	12	10

i	$\epsilon_{i,0}$	$r_{i,0}$	$r_{i,1}$
41	0	9	12
42	1	12	19
43	0	9	20
44	1	3	17
45	0	15	20
46	0	6	14
47	1	19	20
48	0	9	20
49	1	3	12
50	1	2	6
51	1	8	18
52	0	14	5
53	1	17	11
54	1	9	12
55	1	5	16
56	0	12	15
57	1	2	20
58	1	17	9
59	1	16	10
60	1	13	17

i	$\epsilon_{i,0}$	$r_{i,0}$	$r_{i,1}$
61	1	18	17
62	0	14	1
63	0	3	4
64	1	10	17
65	1	7	6
66	1	8	14
67	0	0	10
68	0	8	3
69	0	6	6
70	0	6	6
71	0	18	8
72	1	6	11
73	0	20	19
74	1	17	6
75	1	10	12
76	1	0	11
77	1	19	12
78	0	15	5
79	0	10	4
80	1	3	5

As $\epsilon_{i,1} = 1 - \epsilon_{i,0}$ we only list here $\epsilon_{i,0}$.

The MCQ included $n = 80$ questions. To reduce computational errors, examinees were provided with a form in which they had to report five groups of four numbers. Examinees were instructed to add four consecutive $v_{i,j}$ values[12] using the M+ key and subtract the $v_{i,j}$s again to control that no addition error occurred. If no error occurred, the result would be recalled using the MRC key and copied into the table. In the table, the 20 numbers were divided into five groups of four and added, again, using the same procedure. Finally, the five partial sums were added to get t.

To ease the students' task, a lookup table was also given in the test's appendix. The table gave, for each group of four consecutive questions, sixteen possible sums. Hence – all in all – students could compute t by adding (and controlling the addition of) only 25 integers.

Example: The student's choice (materialized by •s) results in $t = 3355519689$.

The examiner computes:

$$t' = \left(t \times e^{-1} - \left(\sum_{i=1}^{n} a_i \right) \mod p \right) = 388206$$

Hence:

$$q = \left\lfloor \frac{t'}{g} \right\rfloor = \left\lfloor \frac{388206}{9189} \right\rfloor = 42 \quad \text{and} \quad m = \frac{t' - qg}{k} = \frac{388206 - 42 \times 9189}{54} = 42$$

As $0 \leq m \leq n$ and $m \in \mathbb{N}$ we accept $m = 42$ as the student's mark and do not trigger a manual form verification because $\Pr[\ |q - 40| \leq 2] \simeq 0.42$.

B Compressed Answer Encoding

This appendix describes a way to compress IEMCQ answers for $n = 12$. Despite the fact that, in principle, $0 \leq a \leq 12$ and $0 \leq c \leq 12$, we compress the answer into a couple of decimal digits by "reusing" impossible interval notations such as $[7, 3]$.

This is achieved by asking the student to write on the form:

$$\begin{array}{lll} [c-7, a] & \text{if } a \leq 3 & \text{and} \quad c \geq 10 \\ [c-3, a-4] & \text{if } a \geq 4 & \text{and} \quad c \geq 10 \\ [a, c] & \text{otherwise} & \end{array}$$

$n = 12$ is particularly suitable both in terms of answer compactness and standard deviation.

[12] for instance $\texttt{table}_1 = v_{1,0} + v_{2,1} + v_{3,1} + v_{4,0} + v_{5,1}$ *etc.*