Modelling System Checking : An Example of Data Encoding and Traceability

Buda, Rodolphe

Université de Paris 10 Nanterre, GAMA-MODEM

2008

Contrôle des systèmes de modélisation : un exemple de codage et de traçabilité des données

Rodolphe Buda
Economix- UMR 7166 CNRS*
Université de Paris 10

RÉSUMÉ

D'une certaine manière, l'économétrie est l'art de minimiser les erreurs entre une loi quantitative et un échantillon observé. C'est pourquoi, la pratique de la modélisation macro-économétrique nous enseigne qu'en amont du travail économétrique, rien ne doit être laissé au hasard, notamment lors de la construction de la banque de données. Ce papier expose les méthodes classiques de contrôle lors de la collecte et du traitement des données, puis il propose de nouvelles méthodes de codage ("mantisses à dimensions explicites", "mantisses duales" et "mantisses à indices cumulés") permettant la traçabilité des données, à rapprocher de certains algorithmes de la Théorie de l'information.

SUMMARY

We can compare Econometrics with an Art which tries to minimize the lag between a quantitative law and a sample of data. Hence, the econometrics teaches us that before to built models, we must built data bank through strict procedures to prevent errors. Our paper presents classical methods to check data bank building and data computing, then we present new encoding methods ("Explicit Sizes Mantissa", "Dual Mantissa", and "Cumulated Indexes Mantissa") which can lead to follow the data through computing. Some can be based on the Information Theory.

MOTS-CLÉS : Modélisation - Banque de données - Vérification des données - Arithmétique - Algorithmes d'encodage - Mantisse - Chiffres

KEY-WORDS : Modelling - Data bank - Data check - Arithmetic - Encoding Algorithms - Mantissa - Digits

JEL Classification : C81, C82, C87

* @ *rodolphe.buda@u-paris10.fr* - ☎ 01-40-97-77-88 - 🖷 01-47-21-46-89 - ✉ 200, Avenue de la République, 92001 NANTERRE Cedex - FRANCE

"[...] *il y a une différence fondamentale entre les simples données et les observations. Ces dernières* [...] *sont supposées naître d'une observation organisée, guidée par la théorie.* [...] *Les observations sont préparées* [...] *les autres sont simplement obtenues."*
Oskar MORGENSTERN, *L'illusion statistique*, Paris, Dunod, trad.1972, pp.80–81.

La modélisation quantitative ne consiste pas seulement à programmer des procédures de calculs $\boxed{T}$ - traitement des données. Il s'agit également de gérer des flux de données présentant une structure (S_1) en amont et (S_2) en aval dans une banque de données. La fiabilité des résultats peut nécessiter des phases de vérification, voire une systématisation de cette vérification, c'est-à-dire la mise en œuvre de systèmes de contrôle de la séquence

$$(S_1) \rightarrow \boxed{T} \rightarrow (S_2)$$

Ainsi, lorsque la taille des échantillons est faible, la structure des données amont (input) et aval (output) gérée par le programme ne pose quasiment aucun problème. Par exemple dans le cas d'une équation $Y_t = X_t.a + \varepsilon_t$, les données d'input et d'output sont de simples séries temporelles. En revanche, dans le cas d'une équation $Y_t^{r,s} = X_t^{r,s}.a^{r,s} + \varepsilon_t^{r,s}$, la structure des données selon plusieurs les dimensions[1] doit être manipulée avec précaution, faute de quoi on ne peut être sûr ni d'avoir pris en compte les bonnes données, ni d'avoir calculé et/ou stocké correctement les résultats.

Bien qu'il ne s'agisse pas de créer un modèle dans le modèle, une telle systématisation renvoie implicitement aux problématiques du "chiffrement"[2] et plus généralement à celles des théories de la communication et de l'information[3]. Dans des contextes particuliers - militaire, bancaire - la structuration des données doit parfois prévoir le recours à la cryptographie[4] pour sécuriser l'accès et la confidentialité des données.

Après avoir préalablement exposé les règles de base du codage arithmétique de données appliquées aux banques de données de la modélisation, nous illus-

[1]- Notre expérience a porté sur des données multi-périodiques (t), multi-régionales (r), multi-sectorielles (s).

[2]- C'est-à-dire de l'*encodage des réponses aux questionnaires* [14]. A propos de l'automatisation du chiffrement à l'INSÉÉ, voir également [15, 4, 24, 17] ainsi que [2] pour les données textuelles. Pour les sources étrangères, on consultera [29, 16, 26]. Du côté des mathématiciens voir [6].

[3]- La simple opération de saisie constitue déjà une opération de communication. Ajoutons que l'utilisation des scanners et des logiciels de reconnaissance optique des caractères a réduit les phases de traitement manuel, mais ne les a pas totalement supprimées. A propos des problèmes techniques et humains de l'organisation du traitement des données, voir [7]. Citons également [23] à propos du contrôle de la saisie au clavier dans le cadre de l'assistance aux personnes handicapées.

[4]- Citons l'algorithme RSA [25] - voir en annexe 1.

trerons ces règles en présentant les codages triviaux que nous avons dû mettre au point dans le cadre du développement du logiciel `SIMUL` - Système Intégré de Modélisation mULtidimensionnelle. Il s'agit en l'occurrence de codages par données fictives permettant d'assurer la traçabilité des données dans les systèmes de modélisation multi-dimensionnelle. Certains de ces codages nécessitent l'inhibition totale des calculs de la phase de traitement des données, nous parlerons de "*codages passifs*", d'autres autorisent des opérations élémentaires sur les données, nous parlerons alors de "*codages actifs*".

1 - Règles élémentaires du codage de l'information dans une banque de données

Avant de considérer la structure de la banque de données dans le cadre de son utilisation pour simuler ou estimer un modèle, il convient de veiller à construire rigoureusement cette banque. La phase de collecte des données est par conséquent une phase qui nécessite elle même des contrôles. Lors de la collecte des données, il est impératif d'observer les deux règles suivantes[5] : l'*exhaustivité* du champ d'application du codage des données et l'*intégrité* ou exactitude des données stockées[6].

a - L'exhaustivité du système de codage

Le codage de l'information doit pouvoir représenter tous les cas qu'il est nécessaire de coder[7]. S'agissant de données numériques tels que des taux, des effectifs, l'exhaustivité est parfaitement atteinte avec l'arithmétique décimale. Le problème se pose lorsque l'on enregistre des données au moyen de codes, par exemple les réponses à des enquêtes. Il faut alors coder ni plus[8] ni moins (il faut éviter les "impasses"[9]). Cependant dans certains cas, ce principe doit parfois être "aménagé". Lors de l'initialisation des tableaux de données notamment, la mise à zéro systématique des valeurs n'est pas toujours pertinente. Après la saisie des données, les données indisponibles conserveront une valeur nulle ce qui risque en-

[5]- Dans ([21], pp.207–09) R.Reix relève quant à lui cinq règles : l'unicité, l'efficacité à l'identification, la souplesse et perennité, la commodité et la concision ; sans oublier le problème du coût de la fiabilité du système ([22], pp.325–58).

[6]- On pourra consulter ([18],pp.175–97) et ([27], pp.300–62) pour un panorama qualitatif, ainsi que ([11], pp.274–79) au sujet des algorithmes.

[7]- On pourra consulter [1] au sujet des algorithmes optimisés de codage à l'INSÉÉ, en l'occurrence appliqué à l'enquête emploi.

[8]- Le codage des cas improbables est coûteux en place mémoire.

[9]- Un exemple notoire de "codage avec impasse" est celui de l'année de naissance dans les identifiants INSÉÉ des personnes physiques, codée sur deux chiffres. Elle induisait des ambiguïtés puisque "00" correspond à la fois aux millésimes 1900 et 2000.

suite d'induire en erreur lors de l'analyse des résultats. C'est pourquoi on utilise parfois des valeurs notoirement inobservables[10] (par exemple `1.0E+30`).

b - L'intégrité des données

La saisie des données est une opération humaine. A ce titre, elle est entachée d'un risque non nul d'erreur. Deux approches seront abordées. L'une liée au principe de redondance [9], l'autre inhérente à la cohérence du codage des données (c'est-à-dire liée à l'existence d'une relation entre les données).

(i) Vérification de l'intégrité des données et principe de redondance

La procédure qu'il s'agit donc de mettre en œuvre doit donc minimiser cette erreur au moment de la saisie[11]. Une solution triviale de sécurisation consiste à répéter une seconde fois (au moins) l'opération - principe de *redondance*[12]. C'est ce que propose la procédure de "*double-saisie*".

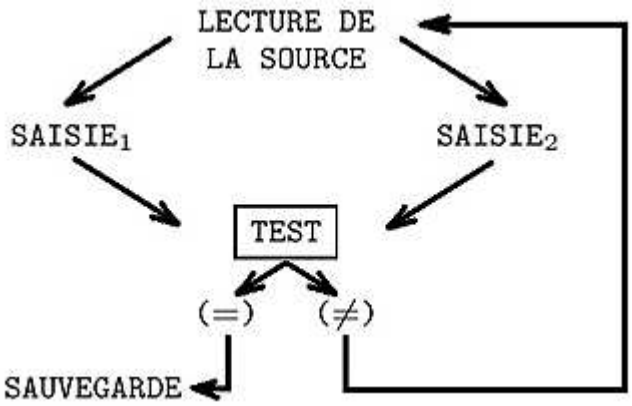

Fig.1 - Procédure de double-saisie

[10]- Le programme doit naturellement prévoir un test sur les données pour éviter d'interrompre les calculs en raison du dépassement de l'arithmétique.

[11]- A titre d'exemple, l'algorithme de chiffrement `QUID`, mis au point par J.Lorigny, décompose en "bigrammes" (couple de lettres) les réponses aux questionnaires. Puis en cheminant selon une arborescence, il compare les principaux bigrammes des mots d'une réponse à une base de données de référence et décide du code correspondant lorsque toute ambiguïté est levée - dans le cas contraire un traitement manuel est alors nécessaire.

[12]- Voir à ce propos ([19], pp.365–80).

Elle consiste à faire saisir par deux personnes différentes[13] le même échantillon dans deux fichiers distincts. Ensuite un programme lit les deux fichiers et compare les données, observation (o^i_j) par observation (où j est l'opérateur et i le rang de l'observation). Deux cas se présentent au rang i - Fig.1.

1° - soit $o^i_1 \neq o^i_2$. On vérifie à la source lequel des deux opérateurs a saisi la bonne observation - il se peut qu'aucun des deux n'ait saisi la bonne ;

2° - soit $o^i_1 = o^i_2$. Dans ce cas aucune vérification n'est faite et le programme de test passe à l'observation suivante[14].

(ii) Vérification de l'intégrité des données et propriétés intrinsèques de l'échantillon

Les informations peuvent être codées de telle manière qu'une simple opération arithmétique permette ensuite d'en vérifier l'intégrité[15]. C'est le cas dans les télécommunications : chaque octet transmis dispose d'un bit appelé bit de parité[16]. Celui-ci permet de contrôler que la somme des autres bits est cohérente avec lui. C'est également le cas avec les numéros de compte bancaire. Chaque numéro de compte dispose d'une clé - voir [5] au sujet des normes internationales. Celle-ci s'obtient en calculant la somme S de tous les chiffres c_i du numéro de compte, puis en prenant le reste R de la division entière de ce nombre par un nombre premier P assez grand[17]. Ce nombre R est appelé la clé[18]

Selon ce principe d'auto-contrôle, de nombreuses séries statistiques sont fournies avec leurs marges (totaux en ligne et/ou totaux en colonne), permettant ensuite de vérifier les données par simples sommations et comparaison aux totaux fournis.

---

[13]- Elle est pour cette raison délaissée eu égard au coût de la saisie.

[14]- La probabilité que deux opérateurs différents se soient trompés de la même manière pour la même observation est très faible - voir annexe 2.

[15]- A propos des codes autodétecteurs d'erreurs et des algorithmes autocorrecteurs d'erreurs ([9], op.cit.)

[16]- Voir ([12], pp.113–46.

[17]- Soit un numéro de compte `6766541`. Si c_i est le i-è chiffre de la mantisse de ce nombre et I le nombre total de chiffres, on a $c_1 = 6$, $c_2 = 7$, etc. Posons $S = \sum_{i=1}^{7} c_i$ on trouve $S = 37$. Si l'on prend $P = 23$, on forme alors $S = Q * P + R$.

$\begin{array}{c|c} S & P \\ \hline R & Q \end{array}$: $\begin{array}{cc} 37 & 23 \\ \hline 14 & 1 \end{array}$: $R = 14$: $\boxed{\text{Clé} = \text{"}N\text{"}}$ "N" étant la 14^e lettre de l'alphabet.

[18]- Si la clé est une lettre, il s'agit de la lettre de rang R dans l'ordre alphabétique. En réalité, nous avons légèrement simplifié la présentation. Le numéro de compte est multiplié par un nombre et la clé de contrôle est souvent 23, plus grand nombre premier inférieur à 26 (nombre de lettres de l'alphabet).

Signalons toutefois que la construction des banques de données de modélisation a bénéficié des progrès de la gestion des bases de données. Notamment leur systématisation - la méthode MERISE - initiée par [28] et qui a permis l'optimisation des procédures de requête[19]. Par ailleurs, s'agissant des logiciels de statistiques, précisons qu'ils permettent de manière systématique une *saisie contrôlée des données*. L'utilisateur doit saisir un nombre restreint de caractères différents et des contrôles de cohérence sont effectués pendant la saisie.

2 - Contrôle de la structure des données après traitement

Nous avons dit que les données en input sont rangées selon une structure (S_1) et qu'à l'issue du traitement $\boxed{T}$, les résultats sont rangés selon une structure (S_2). Malheureusement, la simple lecture des résultats ne permet presque jamais de retrouver cette structure. C'est pourquoi nous avons proposé un codage par des données fictives. Nous verrons que l'utilisation de certains de ces codages nécessite l'inhibition des traitements (les algorithmes de traitement deviennent des algorithmes de transmission), alors que d'autres codages sont compatibles avec des opérations arithmétiques élémentaires.

a - Contrôle des données non calculées : "mantisses à dimensions explicites"

Le codage intuitif que nous proposons consiste à former chaque enregistrement à partir de son rang dans les différentes dimensions d'observation - voir Tab.1 les "mantisses à dimensions explicites"[20].

[19]- Elle a introduit le principe de non redondance des données. Lorsqu'une même donnée existe en plus d'un seul exemplaire dans un système d'information, la mise à jour peut entraîner des erreurs - voir panorama de ces problèmes dans [10].

[20]- Cette technique est assez proche de celle utilisée dans les langages de SGBD - Systèmes de Gestion des Bases de Données - en particulier les techniques de concaténation de champs ([10] *op.cit.*).

Tab.1 - Repérage de la structure par classement

chronologique	régional	sectoriel
10190	10190	10190
10191	20190	10290
10192	30190	10390
10193	40190	10490
⋮	⋮	⋮
10100	220190	13990

Si un échantillon est observé selon les trois dimensions r (régionale), s (sectorielle) et t (temporelle), alors chaque donnée $X_t^{r,s}$ (avec $r \in [1,R]$, $s \in [1,S]$ et $t \in [1,T]$) sera codée sous la forme `rrsstt` - ou n'importe quelle combinaison de dimensions -, c'est-à-dire

$$X_t^{r,s} = r.10^4 + s.10^2 + t$$

b - Contrôle des données calculées

Comme on peut le constater, les données codées avec une mantisse à dimensions explicites ne conservent plus leur propriété de traçabilité lorsqu'elles entrent dans des traitements numériques. Or, il est intéressant de suivre des données à l'issue de certaines opérations numériques particulières, telles que les sommations ou les agrégations (resp.). C'est pourquoi nous avons mis au point le codage avec "*des mantisses duales*" et le codage avec des "*mantisses à indices cumulés*" (resp.).

(i) Calcul de marges : "mantisses duales"

On peut remarquer que si l'on encode les données $X_t^{r,s}$, au format `1.0E+tt`, c'est-à-dire que

$$X_t^{r,s} = 1.0 * 10^t$$

la mantisse comporte alors deux informations : l'une concerne la dimension r (ou s selon le cas) l'autre concerne t. On peut ensuite contrôler - voir Tab.2 - l'ordre

chronologique des opérations de sommation[21]

$$\sum_{r=1}^{R}(1.0*10^{t}) = R*10^{t}$$

Tab.2 - Classement chronologique des sommations

régionales	sectorielles
2.2E+01	3.9E+01
2.2E+02	3.9E+02
2.2E+03	3.9E+03
2.2E+04	3.9E+04
⋮	⋮
2.2E+10	3.9E+10

(ii) Agrégation de données : "mantisses à indices cumulés"

Il s'agit ici de vérifier que les éléments nécessaires (ni plus ni moins) à l'agrégation ont bien été comptabilisés[22]. Soit X un vecteur comportant N observations et Y le vecteur agrégé qui s'obtient comme suit :

$$y_1 = \sum_{i=1}^{p} x_i$$

$$y_2 = \sum_{i=p+1}^{p+q} x_i$$

et ainsi de suite. Pour résoudre ce problème, nous avons codé les données de telle sorte qu'elles soient porteuses de deux informations : la classe d'agrégation à laquelle la donnée appartient et le rang dans la classe. Nous proposons de substituer

[21]- Raisonnement identique avec $\sum_{s=1}^{S}(1.0*10^{t}) = S*10^{t}$.

[22]- Voir [3] à propos de la place et du traitement de l'agrégation dans la modélisation économique.

à la valeur de chaque élément x_i la valeur codée suivante : $\boxed{\texttt{c.0E+i}}$ où c est la classe d'agrégation ($c \leq 9$). D'où le calcul de la première donnée agrégée :

$$y_1 = \sum_{i=0}^{p} x_i = c.10^0 + c.10^1 + c.10^2 + \ldots + c.10^p$$

En généralisant[23], il vient que $y_c = \underbrace{\texttt{cc}\ldots\texttt{c}}_{k}$, où k le nombre d'éléments de X à agréger. On peut alors localiser précisément les erreurs d'agrégation - voir Fig.3 -, grâce à la redondance sur le chiffre c : la mantisse des résultats cumule l'indice d'agrégation autant de fois qu'il y a d'éléments à agréger[24].

$$X = \left(\begin{array}{r} a \\ a0 \\ a00 \\ \hline a000 \\ b \\ b0 \\ \hline \vdots \\ x_N \end{array}\right) : Y = \left(\begin{array}{r} aaaa \\ \hline bb \\ \hline \vdots \end{array}\right)$$

Fig.3 - Traçabilité de l'agrégation

3 - Conclusion

On le voit donc, la construction de banque de données n'est pas une simple formalité. Certains économistes n'hésitent pas à dire que la banque de données du modèle constitue le modèle lui-même. A travers les illustrations de codages de données que nous avons présentées, il nous aura été permis de voir que l'économiste modélisateur est largement aidé par les progrès de l'informatique. Si les capacités de calculs se sont accrues, d'importants progrès sont également apparus en matière de fiabilité de la relation homme-machine[25] ainsi qu'en algorithmique notamment arithmérique[26]. Ce qui permet ainsi de se consacrer plus amplement aux problèmes économiques et économétriques des modèles.

---

[23]- On peut aller au delà de 9 classes en utilisant des codes alphanumériques, mais le programme d'agrégation doit alors être modifié en conséquence.

[24]- Une comparaison analogue à la méthode du décodage syndromique, voir annexe 3.

[25]- Voir [23] en ce qui concerne le contrôle de la saisie au clavier dans le cadre de l'assistance aux personnes handicapées.

[26]- De nouveaux algorithmes sont apparus depuis l'avènement de l'algorithme RSA - voir à ce propos [13]. Voir [8] et [20] au sujet de l'utilisation des algorithmes SHA - Standard Hash Algorithm - permettant de gérer des fichiers de manière anonymisée.

Références

[1] Arrivault G., C.Bergera, M.Cézard & N.Roth, "Les chiffrements automatiques dans l'enquête emploi", *Actes des journées de méthodologie statistique, 17-18 juin 1992*, 1995.

[2] Bruneau E. & P.Riviere, "Recherches textuelles, codage automatique, codage assisté", *Le Courrier des statistiques*, N°81-82, juin, pp.41-47, 1997.

[3] Buda R., "Two Dimensional Aggregation Procedure : An Alternative to The Matrix Algebraic Algorithm", *Computational Economics*, (A paraître).

[4] Dutierez M.C., "Une nouvelle chaîne de traitement", *Le Courrier des statistiques*, N°53, mars, pp.27-35, 1990.

[5] European Committee for Banking Standards, *IBAN : International Bank Account Number*, ECBS, Bruxelles, 2003.

[6] Gennero M.C. & O.Papini, "Utilization of error correcting codes for data transmission simulations", *Discrete Maths.*, N°56(1-2), p.155, 1983.

[7] Gouet M.J., "L'organisation du traitement informatique dans les services régionaux et départementaux de statistique agricole", *Le Courrier des statistiques*, N°32, oct., pp.16-22, 1984.

[8] Goy A., "L'appariement sécurisé de fichiers d'étudiants grâce au hachage des identifiants", *Le Courrier des statistiques*, N°113-114, mars-juin, pp.27-32, 2005.

[9] Hamming R.W., "Error Detecting and Error Correcting Codes", *The Bell System Technical Journal*, apr., N°26(2), pp.147-60, 1950.

[10] Hainaut J.L., *Bases de données et modèles de calcul*, Paris, Dunod, Coll. Science Sup, 435 p., 2005.

[11] Hsu H.P., *Communications analogiques et numériques - Cours et problèmes*, Paris, McGraw-Hill, Coll. Schaum, 330 p., 1994.

[12] Lipschutz M.M & S.Lipschutz, *Traitement de l'information - cours et problèmes*, Paris, McGrawHill, Coll. Schaum, 212 p., 1987.

[13] Le Roux J., *Notions de communication numérique*, Ecole Supérieure en Sciences Informatiques, Nice, 2001.

[14] Lorigny J., "Du nouveau dans l'automatisation du chiffrement à l'INSÉÉ", *Le Courrier des statistiques*, N°20, oct., pp.34-36, 1981.

[15] Lorigny J., "QUID, une méthode générale de chiffrement automatique", *Techniques d'enquête*, déc., 1988.

[16] Lyberg L. & P.Dean, "Automatic Coding of Survey Responses : An International Review", *Conférence des statisticiens européens*, Work Session on Data Editing, Washington, march, 1992.

[17] Meyer R. & P.Riviere, "SICORE, un outil et une méthode pour le chiffrement automatique à l'INSÉÉ", *Document de travail INSÉÉ*, Paris, INSÉÉ, 14 p., 1997.

[18] Peaucelle J.L., *Les systèmes d'information - La représentation*, Paris, PUF, Coll. Systèmes-décision, 249 p., 1981.

[19] Poli A. & H.Huguet, *Codes correcteurs - Théorie et applications*, Paris, Masson, Coll. Logique mathématiques informatique, 448 p., 1989.

[20] Quantin C., B.Gouyon, F.A.Allaert & O.Cohen, "Méthodologie pour le chaînage de données sensibles tout en respectant l'anonymat : application au suivi des informations médicales" *Courrier des statistiques*, N°113-114, mars-juin, pp.15-26, 2005.

[21] Reix R., *L'analyse en informatique de gestion - tome 1, Principes méthodologiques*, Paris, Dunod, Coll. Université et technique, 303 p., 1971.

[22] ————, *Systèmes d'information et management des organisations*, Paris, Vuibert, Coll. Gestion, 372 p., 1995.

[23] Raynal M., *Claviers GAG : claviers logiciels optimisés pour la saisie de texte au stylet*, IHM 2006, Montréal (Canada), 18-21 avr., ACM Press, 2006.

[24] Riviere P., "SICORE, un outil et une méthode pour le chiffrement automatique à l'INSÉÉ", *Le Courrier des statistiques*, N°74, août, pp.65-69, 1995.

[25] Rivest R., A.Shamir & L.Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communication of ACM*, N°21, pp.120-26, 1977.

[26] Schuhl P., "The INSEE Automatic Coding System", *Proceedings of the Annual Research Conference*, Bureau of Census, Washington, 1996.

[27] Senn J.A., *Analyse et conception des systèmes d'information*, Paris, McGrawHill, 647 p., 1986.

[28] Tardieu H., D.Nanci D. & D.Pascot, *Conception d'un système d'information - construction de la base de données*, Paris, Editions d'Organisation, 192 p., 1979.

[29] Wenzowski M., "ACTR, A Generalized Coding System", *Survey Methodology*, dec., 1988.

Annexe I - Confidentialité des communication avec l'algorithme RSA

Deux personnes doivent communiquer entre elles, S qui envoie le message et R qui le reçoit. Lors de la phase de codage, R choisit deux entiers p et q premiers entre eux - *i.e.* n'ayant pas de diviseur commun - et à 200 chiffres environ.

Il calcule

$$n = p.q$$

et

$$\phi(n) = (p-1).(q-1)$$

puis choisit la clé de cryptage

$$e \ / \ PGCD\{e, \phi(n)\} = 1$$

Les valeurs e et n sont communiquées publiquement, mais p, q et $\phi(n)$ sont gardées secrètes.

Le message doit être codé en une suite d'entiers a $a < n$ et a et n premiers entre eux. S envoie alors à M le résultat de l'équation suivante :

$$c = a^e \pmod{n}$$

Lors de la phase de décodage, R calcule la clé de décryptage en résolvant l'équation en d suivante :

$$e.d \equiv 1 \ mod \ \phi(n)$$

avec

$$1 < d < \phi(n)$$

Grâce au Théorème d'Euler, appelé aussi "*petit théorème de Fermat*", on obtient la simplification :

$$c^d = a \pmod{n}$$

Annexe 2 - Minimisation des erreurs de saisie avec la procédure de la double-saisie

Supposons que les deux opérateurs saisissent par erreur le même nombre N_0 comportant n chiffres

$$N_0 = [c_0^1][c_0^2]\dots[c_0^n]$$

Si $\phi_j^{0,i}$ la probabilité pour l'opérateur j de saisir au $i$ème rang le chiffre c_0^i, avec $0 \leq \phi_j^{0,i} \leq 1$, alors la probabilité pour l'opérateur j de saisir le nombre N_0 est donc

$$P_j^0 = \prod_{i=1}^{n} \phi_j^{0,i}$$

d'où la probabilité $P^0_{1\ \&\ 2}$ que les deux opérateurs aient fait la même erreur (saisie du même nombre N_0) est

$$P^0_{1\ \&\ 2} = \prod_{j=1}^{2} P_j^0$$

$$=: P^0_{1\ \&\ 2} = \prod_{j=1}^{2} \left(\prod_{i=1}^{n} \phi_j^{0,i} \right)$$

on a alors

$$\boxed{P^0_{1\ \&\ 2} \ll \inf_j P_j^0}$$

c'est-à-dire que la probabilité que les deux opérateurs se trompent simultanément de la même manière (N_0), est très inférieure à la probabilité que le plus vigilant des deux opérateurs fasse précisément l'erreur N_0.

De plus, même si dans la pratique cela ne présente pas d'intérêt[27], on montre facilement que

$$\lim_{n \to +\infty} P^0_{1\ \&\ 2}(n) \to 0$$

et

$$\lim_{j \to +\infty} P^0_{1\ \&\ 2}(j) \to 0$$

[27]- La mantisse (n) des nombres n'excède pas une douzaine de chiffres, de même que le nombre d'opérateurs (j) de saisie n'excède pas deux.

Annexe 3 - Contrôle des erreurs par la méthode du décodage syndromique[28]

Les théories de la communication et de l'information nous enseignent que l'on peut former le codage c d'un message en utilisant des blocs séquentiels[29] :

$$\underbrace{c_1 c_2 \dots c_k}_{\textit{Bits de données}} \quad \underbrace{c_{k+1} c_{k+2} c_n}_{\textit{Bits de parité}}$$

Les k premiers chiffres sont les bits de données tandis que les $n-k$ suivants sont les bits de contrôle. Si c est un *vecteur code* tel que $c = \left(c_1 \dots c_n\right)$, et d un *vecteur données* tel que $d = \left(d_1 \dots d_k\right)$, alors, le *codage systématique* consiste à former :

$$c = dG$$

avec

$$G = \left[I_k P^T\right]$$

et

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1k} \\ p_{21} & p_{22} & \cdots & p_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mk} \end{pmatrix}$$

on forme la matrice de parité H :

$$H = \left[P I_m\right]$$

d'où sa transposée

$$H^T = \begin{bmatrix} P^T \\ I_m \end{bmatrix}$$

on voit alors que

$$GH^T = [I_k P^T] \begin{bmatrix} P^T \\ I_m \end{bmatrix}$$

$$: GH^T = P^T \oplus P^T$$

$$: GH^T = 0_{(k,n-k)}$$

Si r est un vecteur reçu en transmission et e le vecteur erreur tel que

$$\begin{cases} e_i = 1 \text{ et} \\ e_k = 0 \; \forall \; k \neq i \end{cases}$$

le contrôle des erreurs de transmission s'opère en évaluant $r.H^T$:

$$r.H^T = (c \oplus e).H^T$$

$$r.H^T = 0 \oplus e.H^T$$

$$r.H^T = e.H^T$$

$$r.H^T = s$$

ce qui permet de repérer la position d'une erreur en comparant s aux lignes de H^T puisque $e.H^T$ est la iè ligne de H^T.

[28]- D'après ([11], pp.274-79).

[29]- Tous les chiffres sont en binaire, et les opérations, *modulo* 2.