2) Le protocole est cohérent car on peut toujours donner une réponse au défi. Dans le protocole, on peut anticiper le défi 0 donc la probabilité de tromperie est au moins $1/q$. On ne peut anticiper plus d'un défi.
3) Si l'on prend $q = 2^k$, on obtient une sécurité en un tour quand il en faudrait k pour Fiat-Shamir.

8.9. 1) On choisit au départ un paramètre de sécurité en 2^{-k}. On choisit k engagements distincts comme si l'on allait faire k tours. Ces engagements sont concaténés au message à signer. On hache alors le tout et l'on en extrait k défis. On calcule les réponses associées aux défis. La signature est le message plus les engagements prédéfinis ainsi que les réponses. Pour vérifier la signature, on vérifie que les défis ont été calculés convenablement et l'on vérifie le protocole. Comme les défis sont choisis aléatoirement (à cause de la fonction de hachage) après le choix des engagements, on est sûr qu'ils n'ont pas été anticipés.
2) Pour une sécurité en 2^{80}, il faut prendre 160 bits de défi pour éviter les attaques par paradoxe des anniversaires, soit $k = 160$. Puisqu'en un tour il y a 2.1024 bits échangés en tout, cela donne une signature de 320 kilobits.

8.10. Soit la valeur secrète r. On choisit un corps fini tel que le secret soit de la taille d'un élément du corps. Par exemple, un secret de 128 bits donne le corps $K = \mathbb{F}_{2^{128}}$. Pour un secret retrouvable par au moins k personnes parmi n, on choisit un polynôme f sur $K[x]$ de degré k tel que $f(0) = r$ (il suffit de fixer la constante de f à r). On distribue alors aux n personnes une valeur $f(a_i)$, pour des a_i distincts et non nuls. Si un groupe de k personnes collabore, elles sont capables de reconstruire f par interpolation (voir le chapitre 2) car f est de degré k. Si moins de k personnes collaborent, elles retrouveront un polynôme à une indéterminée près. Comme le secret est justement une constante, elles ne peuvent retrouver d'informations supplémentaires sur le secret.

8.11. 1) A choisit une clé de session k puis la casse en deux morceaux k_1 et k_2. Elle envoie un message commençant par $E_{K_{AB}}(k_1)$ et $E_{K_{AC}}(k_2)$, puis $E_k(M)$. Pour déchiffrer, B et C auront à collaborer pour reconstruire k à partir de k_1 et k_2.
2) On écrit cette fois $k = k_1 + k_2 + k_3$ et l'on envoie, chiffrés par les clés symétriques, le couple $\{k_1, k_2\}$ à B, le couple $\{k_2, k_3\}$ à C et le couple $\{k_1, k_3\}$ à D avant d'envoyer le message M chiffré par la clé de session k. Si deux parmi trois collaborent, ils peuvent retrouver k_1, k_2 et k_3 puis retrouver k, mais un seul ne le peut pas.
3) On peut soit généraliser un point de vue combinatoire comme au 2). soit appliquer le schéma de partage de secret de Shamir de l'exercice précédent.

8.12. Lorsque l'on chiffre, on calcule $mG' + e$. Si G' est sous forme systématique, comme l'erreur est proportionnellement faible, on peut retrouver des informations sur les premiers bits de m. Pour pouvoir quand même utiliser une forme systématique, on fixe e et, au lieu de chiffrer m, on chiffre $m + h(e)$ pour une fonction h de hachage. Pour déchiffrer, on trouve l'erreur e et l'on décode $m + h(e)$. On peut donc retrouver $m = m + h(e) - h(e)$.

8.13. 1) Pour t petit, le nombre de mots dans une boule de rayon t sur $\mathbb{F}_2^n$ est très proche de $\binom{n}{t}$ (dans notre cas $n = 2^m$). La densité des mots décodables est donc le nombre de mots du code fois le nombre de mots dans la boule divisé par le nombre de mots de l'espace, soit $\frac{\binom{n}{t} 2^{m-mt}}{2^m}$. On peut approcher $\binom{n}{t}$ par $\frac{n^t}{t!}$ ce qui, en remplaçant dans l'égalité de la densité, donne une densité en $1/t!$.
2) La question 1) nous dit que si l'on prend un mot au hasard dans l'espace, il y a en moyenne une chance sur $t!$ qu'il soit décodable. On fait donc une signature qui fonctionne comme une fonction *pseudo-inversible* de type R.S.A. (mais pour R.S.A., la signature vient de l'inversabilité de la fonction de chiffrement). On reprend les mêmes clés publiques/privées que pour le schéma de McEliece. À partir du message à signer m, on peut donc mettre en place un compteur et décoder un mot de l'espace $\mathbb{F}_2^m$ dérivant de $h(m \oplus i)$ (pour une fonction h de hachage) en augmentant i jusqu'à ce que l'on obtienne un mot décodable (en moyenne au bout de $t!$ essais). Le secret est donc le fait d'être capable de décoder G'. Mais comme on ne sait le faire que pour certains mots de l'espace, on dérive des mots aléatoires de l'espace jusqu'à pouvoir les décoder. La signature est donc le mot de l'espace $x = h(m \oplus i_0)$ et son décodage par la matrice G'. Pour vérifier, on vérifie que $x = h(m \oplus i_0)$ et que le mot décodé est bien dans le code.
3) Comme il faut faire $t!$ décodages en moyenne avant de trouver le bon mot, on ne peut pas prendre t trop grand. En même temps, pour t petit, on peut attaquer plus facilement. Si l'on se fonde sur l'attaque décrite dans le cours, on obtient $m \geqslant 16$. En stockant uniquement la matrice duale, on obtient une clé d'au moins 8 mégabits. C'est beaucoup ! Mais cela donne un schéma de signature avec les codes.

Chapitre 9

9.1. 1) Si l'on note $f(x, y) = x^2 + 2xy + 2y^2 + \sqrt{15}$, il est clair que cette fonction est une forme quadratique de la forme $f(x, y) = \frac{1}{2}\langle A.(x, y), (x, y)\rangle + \sqrt{15}$ avec

$$A = \begin{pmatrix} 2 & 2 \\ 2 & 4 \end{pmatrix}.$$

De plus, les valeurs propres de cette matrice sont $\lambda_1 = 3 - \sqrt{5}$ et $\lambda_1 = 3 + \sqrt{5}$, qui sont toutes les deux positives. La forme quadratique f est donc convexe.