1.1 Notions de sécurité pour le chiffrement asymétrique

1.1.1 Sécurité parfaite et sécurité au sens de la complexité

La première étape dans l'évaluation de la sécurité est, bien évidemment, la compréhension de la notion de « sécurité ». À ce jour, on en connaît deux définitions majeures.

La première, au sens de la théorie de l'information, a été initialement évoquée par Shannon [113]. Elle concerne l'« information » sur le texte clair « contenue » dans le texte chiffré : un schéma de chiffrement est *parfaitement sûr* si le texte chiffré ne contient aucune information au sujet du texte clair correspondant. Dans le cadre d'un schéma de chiffrement symétrique, il a été démontré que la sécurité parfaite n'est atteinte que si la clef utilisée est aussi longue que le message. Cette condition est une sérieuse limite en pratique.

La deuxième approche, au sens de la théorie de la complexité, est actuellement utilisée dans l'analyse des schémas de chiffrement. Elle ne s'intéresse pas au contenu du texte chiffré mais met l'accent sur la « difficulté » d'extraire de l'information sur le texte clair à partir du texte chiffré. Cette difficulté est considérée au sens de la complexité et elle est souvent comparée à la difficulté d'un problème bien défini : la factorisation, par exemple.

Remarquons que la notion de « sécurité parfaite » n'est pas applicable au chiffrement asymétrique car le texte chiffré contient toujours de l'information sur le texte clair. En effet, tout attaquant de puissance illimitée peut retrouver le texte clair à partir du texte chiffré grâce à une recherche exhaustive dans l'espace des textes clairs et éventuellement, dans un espace d'aléas (dans le cas d'un chiffrement probabiliste). Dans ce qui suit, nous présentons les différentes notions de sécurité d'un schéma de chiffrement qui sont caractérisées par deux éléments : la difficulté d'extraire de l'information et la puissance dont un attaquant dispose pour casser le schéma.

1.1.2 Réduction en terme de complexité

Dans les premiers schémas de chiffrement à clef publique comme RSA [107] ou Merkle-Hellman [81], la nature de la sécurité n'a pas été prouvée, faute d'une définition de la sécurité. En 1979, Rabin [103] a introduit un schéma où la capacité d'un attaquant d'extraire le message complet à partir d'un texte chiffré est calculatoirement équivalente à la factorisation. Le schéma de chiffrement de Rabin est décrit comme suit : chaque utilisateur choisit deux nombres premiers de même taille p, q (servant de clef secrète) et publie la clef publique $n = pq$. Le chiffrement d'un message m est $c = m^2 \bmod n$. Le receveur, grâce à la connaissance de p, q, peut facilement calculer les quatre racines carrées du chiffrement c et en choisir le message approprié (une façon pratique de conduire à un bon choix est de mettre de la redondance dans le message initial). La sécurité considérée pour ce schéma est dite *à sens-unique* (dénotée OW, par *one-way* en anglais). Un schéma est « à sens-unique » si, à partir d'un *challenge* chiffré, l'attaquant ne peut retrouver le texte

clair correspondant.

Pour prouver la sécurité, on effectue une *réduction* d'un *problème algorithmique* à un *problème de sécurité*. Dans ce schéma, les deux problèmes sont respectivement une solution de la factorisation et l'extraction du message complet à partir d'un texte chiffré. En entrée, une instance n du problème de factorisation est donnée (n est la multiplication de deux premiers). Le but est de résoudre le problème de factorisation pour l'instance n en utilisant un attaquant contre le schéma. L'attaquant, étant capable d'extraire le message complet à partir d'un texte chiffré, joue le rôle de *boîte noire*, *i.e.* il reçoit une entrée et en retourne une sortie sans que l'on ne connaisse le mécanisme. Pour utiliser cet attaquant, il nous faut donc construire un *simulateur* de l'environnement d'attaque. Le simulateur créera un environnement parfaitement similaire (ou au moins indistinguable aux yeux de l'attaquant) à l'environnement réel de l'attaque.

Dans le cas du chiffrement de Rabin, un tel simulateur peut être construit de façon simple. En effet, il publie le nombre n comme la clef publique, puis, choisit un texte aléatoire m et le chiffre comme $c = m^2 \bmod n$. Le simulateur donne la clef publique et le chiffré c à l'attaquant. Comme il s'agit d'un attaquant passif (qui ne demande pas d'information supplémentaire au système), le simulateur n'a pas à simuler les informations à échanger. La simulation est alors parfaite. L'attaquant retournera les quatre racines carrées correspondant au chiffré c. Le simulateur choisit m' parmi ces quatre racines carrées tel que $m' \neq \pm m \bmod n$. Alors, un des deux facteurs de n est le plus petit diviseur commun de n et $m - m'$, d'où la factorisation de n.

Le schéma de Rabin est ainsi le premier système dans lequel une *hypothèse algorithmique* — la difficulté de la factorisation au sens de la complexité — a été réduite à un problème de sécurité — la capacité d'extraire le message complet à partir d'un texte chiffré. Une telle réduction s'appelle aujourd'hui « une preuve de sécurité ».

1.1.3 Fonctions à sens-unique

Dans ce premier exemple, on peut traduire l'« hypothèse algorithmique » de la difficulté de la factorisation en propriété « à sens-unique » de la fonction de multiplication de deux nombres premiers, *i.e.* elle est facile à calculer mais difficile à inverser. Cette notion de *fonction à sens-unique* est fondamentale en cryptographie. Dans les schémas asymétriques, elle est souvent couplée avec une propriété supplémentaire : l'inverse peut être facilement calculé grâce à quelques informations additionnelles, *i.e.* une trappe. Un des candidats pour les permutations à sens-unique à trappe est la fonction RSA où l'inversion est conjecturée difficile mais devient facile avec la connaissance de la factorisation du module utilisé. Une permutation à sens-unique à trappe implique naturellement un schéma OW contre les attaquants passifs. Cependant, la recherche en cryptographie considère d'autres notions de sécurité plus subtiles que la propriété à sens-unique et d'autres types d'attaquants plus puissants qu'un attaquant passif. À ce stade, une question importante est la suivante : « la cryptographie peut-elle être fondée uniquement sur les hypothèses générales telles que l'existence de fonctions (permutations) à sens-unique ou de fonctions

(permutations) à sens-unique à trappe ? ».

1.1.4 Sécurité sémantique et indistinguabilité

Dans les années 80, les chercheurs ont formellement défini les notions de sécurité pour les primitives cryptographiques (notamment pour la signature [60, 61] et pour le chiffrement [59]). Goldwasser et Micali [59] ont introduit la notion de la *sécurité sémantique* et ont proposé la construction d'un chiffrement probabiliste qui peut garantir la sécurité sémantique.

La notion de sécurité sémantique est une notion très forte, elle couvre l'exigence de ne pas pouvoir extraire une information, même partielle (ne serait-ce qu'un seul bit) sur le clair à partir du chiffré. Elle coïncide avec la notion de la sécurité parfaite limitée aux attaques polynomiales probabilistes. La sécurité sémantique est la propriété désirée pour tout schéma de chiffrement, mais, en réalité, on étudie souvent la notion de *l'indistinguabilité* (dénotée IND), plus simple à analyser. Cette notion a également été introduite dans [59]. Goldwasser et Micali [59] (l'indistinguabilité implique la sécurité sémantique) et Micali, Rackoff et Sloan [82] (la sécurité sémantique implique l'indistinguabilité) ont démontré qu'elle était équivalente à la notion de la sécurité sémantique. Avec l'indistinguabilité, l'attaquant ne peut pas trouver deux textes clairs m_0 et m_1 dont il pourrait distinguer les chiffrements : il reçoit un challenge c, chiffré de m_0 ou m_1 et ne doit pas pouvoir deviner à quel texte clair c correspond.

Cette condition implique immédiatement que le chiffrement soit probabiliste. De ce fait, les chiffrements déterministes de RSA et de Rabin n'atteignent pas la sécurité sémantique. Les premiers schémas qui ont atteint cette notion forte sont le schéma de Goldwasser et Micali [59] qui repose sur la difficulté du problème de la résiduosité quadratique, et le schéma de Yao [120], fondé sur l'hypothèse générale de l'existence de fonctions injectives à sens-unique à trappe. Cependant, ces deux schémas sont totalement inefficaces car leurs chiffrements sont « bit par bit », *i.e.* chaque bit est chiffré de manière indépendante, elles conduisent à des chiffrés très larges. Un schéma plus efficace (à chiffrés plus courts) ayant atteint la propriété d'indistinguabilité est celui de Blum et Goldwasser [16]. Il est fondé sur la difficulté du problème de la factorisation. L'idée dans ce schéma est d'utiliser le générateur Blum-Blum-Shub [14] pour générer des bits aléatoires qui masqueront ensuite le clair.

1.1.5 Attaques à chiffrés choisis

Ainsi, l'indistinguabilité a été initialement définie dans le scénario de base où l'attaquant n'a accès qu'à l'information publique et, de ce fait, peut chiffrer des clairs de son choix, d'où le nom d'« attaques à clairs choisis » (dénotée CPA, par *chosen plaintext attack* en anglais). Désormais, on dénote un schéma qui est sûr au sens XXX (IND, par exemple) face aux attaques YYY (CPA, par exemple), un schéma $\mathsf{XXX}-\mathsf{YYY}$ sûr ($\mathsf{IND}-\mathsf{CPA}$ sûr,

par exemple).

Naor et Yung [87] ont étudié la notion d'attaque à chiffrés choisis dans le cas où l'attaquant peut accéder à un oracle (dit l'oracle de déchiffrement) qui retourne, pour chaque chiffré, le clair correspondant.

Face à la puissance de ce type d'attaque, tous les schémas précédents deviennent vulnérables. Naor et Yung [87] ont construit le premier schéma résistant à ce type d'attaque en exploitant la notion de preuve à divulgation de connaissance nulle d'appartenance à un langage (dénotée preuve NIZK, par *non-interactive zero-knowledge proof* en anglais), introduite par Blum, Feldman et Micali [15] (*i.e.* la version non-interactive de la notion de preuve zero-knowledge de l'appartenance à un langage introduite par [58]). Dans [15], Blum *et. al* ont déjà considéré des attaques à chiffrés choisis. Ils ont même suggéré une manière de construire un schéma résistant aux attaques à chiffrés choisis.

Blum *et. al* [15] ont proposé le schéma (sans construction formelle) suivant : au lieu d'envoyer le chiffré $c = E(m)$, il faut envoyer c et σ, une preuve non-interactive zero-knowledge, justifiant que le déchiffrement de c est connu de l'expéditeur. Bien que la notion de NIZK soit proposée dans le même article, remarquons qu'il s'agit plutôt d'une preuve de *connaissance* dans la construction. Le déchiffrement vérifie d'abord si σ est convaincante, auquel cas il retourne m, sinon, il ne retourne rien. De cette façon, l'oracle de déchiffrement semble inutile : pour que l'oracle retourne le message, l'attaquant doit soumettre un chiffré avec une preuve convaincante de connaissance du déchiffrement, *i.e.* il doit déjà connaître le clair. Ainsi, une fois que le chiffrement E est sémantiquement sûr face aux attaques passives, le chiffrement modifié l'est également face aux attaques à chiffrés choisis. Cependant, Blum *et. al* n'ont pas expliqué les étapes de construction d'une telle preuve de connaissance et n'ont pas formellement prouvé la sécurité.

Le schéma proposé par Naor et Yung s'est inspiré de l'idée de Blum *et. al* avec une utilisation judicieuse des preuves NIZK. Le point de départ était toujours un chiffrement E $\mathsf{IND-CPA}$ sûr. Le chiffrement d'un message m retourne un couple $(c_1 = E(e_1, m), c_2 = E(e_2, m))$ avec deux clefs publiques différentes e_1, e_2 (les deux clefs secrètes correspondantes sont d_1, d_2) ainsi qu'une preuve NIZK σ que c_1 et c_2 sont deux chiffrés du même message. Le déchiffrement ne retourne le message que si la preuve est convaincante. L'idée est que la connaissance d'une des deux clefs secrètes d_1, d_2 est suffisante pour le déchiffrement. Alors, le simulateur, en choisissant une de ces deux clefs, disons d_2, pourra bien simuler le déchiffrement et réduire une attaque à chiffrés choisis contre le nouveau schéma à une attaque passive contre $E(e_1)$. Le schéma résiste ainsi aux attaques à chiffrés choisis. Cependant, le modèle d'attaque considéré par Naor et Yung est *à chiffrés choisis non-adaptative* (dénoté $\mathsf{CCA1}$) au sens où les requêtes ne doivent pas dépendre du challenge. En effet, si l'attaquant peut faire des requêtes qui dépendent du challenge, la construction de Naor et Yung ne résiste plus aux attaques à chiffrés choisis. À partir d'une preuve NIZK, il est éventuellement possible de construire une autre preuve NIZK qui ne tient pas compte du dernier bit. Dans ce cas, si (c_1, c_2, σ) est le challenge, alors l'attaquant peut soumettre une requête sur (c_1, c_2, σ'), où σ' ne diffère de σ que par le dernier bit, pour révéler le clair correspondant au challenge. L'attaquant casse ainsi le schéma par

une seule requête adaptative (par rapport au challenge) de déchiffrement. Le schéma de Naor et Yung joue cependant un rôle très important en théorie car c'est le premier schéma à considérer des attaques à chiffrés choisis. De plus, il ne repose que sur une hypothèse générale. En effet, puisque l'existence d'une permutation à sens-unique à trappe est suffisante pour construire un schéma $\mathsf{IND}-\mathsf{CPA}$ sûr [59] et une preuve NIZK [48], elle est aussi suffisante pour le schéma $\mathsf{IND}-\mathsf{CCA1}$ sûr de Naor et Yung.

Le modèle d'attaque *à chiffrés choisis adaptative* (dénoté $\mathsf{CCA2}$) a été étudié pour la première fois par Rackoff et Simon [104]. Dans ce modèle, l'attaquant peut soumettre des requêtes à l'oracle de déchiffrement à tout instant, avant et après la réception du challenge, avec pour seule restriction de ne pas soumettre de requête sur le challenge. Rackoff et Simon ont formalisé la notion de preuve non-interactive zero-knowledge de connaissance, *i.e.* la version non-interactive de la notion de preuve zero-knowledge de connaissance introduite par [46, 47]. En examinant cette dernière, Rackoff et Simon ont proposé un schéma $\mathsf{IND}-\mathsf{CCA2}$ sûr, fondé sur l'hypothèse générale de l'existence de permutations à sens-unique à trappe. Cependant, chaque expéditeur et chaque destinataire dans ce schéma doivent faire confiance à un centre - un troisième participant qui génère et leur distribue la clef publique ainsi que la clef secrète. Par conséquent, ce schéma ne rentre pas dans la catégorie classique des schémas de chiffrement à clef publique [41].

1.1.6 Non-malléabilité

La construction d'un schéma $\mathsf{IND}-\mathsf{CCA2}$ sûr est réalisée grâce à une nouvelle notion de sécurité : la *non-malléabilité* (dénotée NM). Introduite par Dolev, Dwork et Naor [43, 44], elle est considérée comme une extension de la sécurité sémantique (une autre notion de non-malléabilité équivalente et plus simple à étudier sera ultérieurement introduite par Bellare, Desai, Pointcheval et Rogaway [7]), elle est en général plus forte que l'indistinguabilité. Bellare *et. al* [7] ont pourtant prouvé que la non-malléabilité et l'indistinguabilité sont équivalentes face aux attaques à chiffrés choisis adaptatives.

La non-malléabilité exclut les attaques qui, en possession d'un chiffré, peuvent produire un nouveau chiffré tel que les clairs correspondants sont clairement reliés sans pour autant les connaître. Dolev *et. al* [43, 44] ont construit un schéma complexe qui considère l'interaction entre plusieurs chiffrements et repose sur des preuves NIZK ainsi que des signatures à usage unique (« one-time signature » en anglais). Ce schéma est prouvé non-malléable contre des attaques à chiffrés choisis adaptatives. Leur schéma est donc le premier schéma $\mathsf{IND}-\mathsf{CCA2}$ fondé sur l'hypothèse générale de l'existence de permutations à sens-unique à trappe.

Une autre construction, beaucoup plus simple, se fondant sur la même hypothèse, est due à Sahai. Avec une nouvelle notion de preuve non-malléable NIZK [109], Sahai a montré qu'une telle preuve peut être déduite d'une preuve NIZK. Il montrera ensuite qu'avec le remplacement de la preuve NIZK σ par une preuve non-malléable NIZK σ' dans le schéma $\mathsf{IND}-\mathsf{CCA1}$ de Naor et Yung, on peut rendre ce schéma résistant aux attaques à chiffrés choisis adaptatives.

À ce jour, le schéma à clef publique le plus simple et le plus efficace (dans la catégorie classique des schémas de chiffrement à clef publique) qui garantit la sécurité sémantique contre des attaques à chiffrés choisis adaptatives (et donc $\mathsf{NM} - \mathsf{CCA2}$)est le schéma de Cramer-Shoup [37]. Il s'agit du seul schéma qui ne repose pas sur la construction complexe des preuves NIZK. Cependant, d'un point de vue théorique, ce schéma est moins attractif car il ne se fonde pas sur une hypothèse générale mais sur la difficulté d'un problème décisionnel spécifique, *i.e.* le problème Diffie-Hellman Décisionnel, ou tout du moins dans des structures de groupe [39].

1.2 Notions de sécurité pour le chiffrement symétrique

1.2.1 Fonctions pseudo-aléatoires

À la différence de la cryptographie asymétrique, dans la cryptographie symétrique, le chiffrement et le déchiffrement utilisent une même clef secrète. Par conséquent, non seulement le déchiffrement mais aussi le chiffrement sont difficiles à calculer sans la connaissance de la clef. C'est pour cette raison qu'un accès à l'oracle de chiffrement peut donner un certain avantage aux attaquants. C'est pourquoi la cryptographie symétrique considère des attaques à clairs choisis. L'exigence de sécurité est que : sans la connaissance de la clef de chiffrement, la sortie est similaire à une suite de bits aléatoires même si l'on peut tester le chiffrement sur plusieurs couples clair-chiffré. Autrement dit, la sortie est pseudo-aléatoire au sens où il n'y a pas d'algorithme faisable (*i.e.* un attaquant en temps polynomial) qui puisse la distinguer d'une suite de bits véritablement aléatoires.

1.2.2 Permutations (super) pseudo-aléatoires

Pour la catégorie de chiffrement la plus fréquemment étudiée en cryptographie symétrique — le chiffrement par bloc — la propriété désirée est que le chiffrement soit une permutation (super) pseudo-aléatoire. Une permutation pseudo-aléatoire est une fonction pseudo-aléatoire où la fonction est une permutation. La notion de permutation super pseudo-aléatoire est une extension de permutation pseudo-aléatoire où l'attaquant peut accéder non seulement à la permutation mais aussi à son inverse. Cette notion modélise la propriété la plus forte concernant les chiffrements par bloc. Ces derniers seront, à leur tour, utilisés comme primitives pour d'autres applications (chiffrement de données à travers des modes d'opération, code d'authentification de messages,...). Une question générale est de savoir s'il existe des constructions de fonctions pseudo-aléatoires ou permutations (super) pseudo-aléatoires ; et si oui, sous quelle hypothèse.

Goldreich, Goldwasser et Micali [56] ont présenté une construction simple de fonctions pseudo-aléatoires à partir d'un générateur pseudo-aléatoire de longueur double (*i.e.* de n bits à $2n$ bits). À la différence des fonctions pseudo-aléatoires, un générateur pseudo-aléatoire donne une suite de bits pseudo-aléatoires à partir d'une suite plus courte de

bits vraiment aléatoires. Blum et Micali [17, 18], en utilisant un *prédicat difficile* [1] de cette fonction, ont construit un générateur pseudo-aléatoire à partir d'une permutation à sens-unique de façon très efficace. Hastad, Impagliazzo, Levin et Luby ont montré dans [64] (qui est une combinaison des résultats de [65, 63]) qu'un générateur pseudo-aléatoire existe si et seulement si une fonction à sens-unique existe. Bien que leur construction ne soit pas efficace et n'ait jamais été utilisée en pratique, c'est un résultat important d'un point de vue théorique.

Luby et Rackoff [78] ont construit des permutations super pseudo-aléatoires à partir de fonctions pseudo-aléatoires. La construction comprend une construction de Feistel à 4 tours (3 tours pour des permutations pseudo-aléatoires). La réponse concernant l'existence des permutations super pseudo-aléatoires est ainsi affirmative sous la seule hypothèse générale de l'existence de fonctions à sens-unique.

1.3 Discussion

1.3.1 Du point de vue théorique

Reprenons la question « la cryptographie peut-elle être uniquement fondée sur les hypothèses générales ? ». Dans ce travail, nous étudions le cas du chiffrement. Nous avons vu dans les sections précédentes que, pour le chiffrement asymétrique, l'existence de permutations à sens-unique à trappe suffit pour construire des schémas $\mathsf{IND}-\mathsf{CCA2}$ sûrs et que, pour le chiffrement symétrique, l'existence de fonctions à sens-unique suffit pour construire des permutations super pseudo-aléatoires.

Les fonctions à sens-unique sont-elles suffisantes pour le chiffrement asymétrique ? Impagliazzo et Rudich [66] ont partiellement répondu négativement à cette question en prouvant que ces fonctions (voire les permutations à sens-unique comme indiqué après par [70]) ne permettent pas de construction générique d'un schéma $\mathsf{IND}-\mathsf{CPA}$ sûr. Cela signifie que l'on ne peut utiliser les fonctions à sens-unique comme une « boîte noire » pour construire des schémas de chiffrement asymétrique. Cependant, il ne s'agit pas d'une impossibilité absolue ; une construction spécifique et concrètes d'un schéma de chiffrement à clef publique pourra être faisable à partir de fonctions à sens-unique.

Alors que la condition minimale pour chiffrement symétrique est évidemment l'existence de fonctions à sens-unique, celle pour chiffrement asymétrique n'a pas encore été trouvée. Même si on se contente de la notion $\mathsf{OW}-\mathsf{CPA}$, on ne sait pas construire de schéma de chiffrement à partir de fonctions à sens-unique sans trappe. En effet, on remarque d'abord que l'existence des schémas $\mathsf{OW}-\mathsf{CPA}$ sûrs est équivalente à l'existence

[1] Un prédicat difficile B d'une fonction f est un prédicat booléen tel que $B(x)$ est efficacement calculé, étant donné x mais il est difficilement calculé, étant donné juste $f(x)$. Il y a une manière simple de modifier une fonction à sens-unique en une autre fonction à sens-unique telle qu'elle possède un prédicat difficile de la dernière [57].

d'une famille de *prédicats à trappe* [2] [59]. Bellare *et. al* [8] ont montré que les fonctions à sens-unique à trappe, qui sont non injectives et dont la taille des pré-images correspondant à une image est d'ordre polynomial, suffisent pour construire des prédicats à trappe. Dans un effort de construire ces fonctions, Bellare *et. al* ont pu obtenir, à partir des fonctions à sens-unique, des fonctions à sens-unique à trappe non injectives dont la taille des pré-images correspondant à une image est d'ordre super-polynomial. Bien qu'elles ne servent pas à construire des schéma $\mathsf{OW}-\mathsf{CPA}$ sûrs, il s'agit du premier travail ayant pour but d'éviter « la trappe » dans la construction des schéma de chiffrement asymétrique.

1.3.2 Du point de vue pratique

En pratique, on doit trouver un compromis entre l'efficacité (la complexité) du schéma et son niveau de sécurité en faisant quelques hypothèses sur l'attaque. À titre d'exemple, l'attaque pourrait être générique et indépendante de l'exécution réelle de certains objets tels que les fonctions de hachage (dans le modèle de l'oracle aléatoire [49, 9]), le chiffrement symétrique par bloc (dans le modèle du « chiffrement idéal ») ou les groupes algébriques (dans le modèle générique [27]). Parmi ces hypothèses idéalistes, celle sur les fonctions de hachage — le modèle de l'oracle aléatoire — est la plus étudiée. Dans ce « modèle de l'oracle aléatoire », la fonction de hachage est formalisée comme un oracle qui retourne une valeur parfaitement aléatoire. On a mis au point des schémas particulièrement efficaces dans ce modèle comme OAEP pour le chiffrement ou PSS (*Probabilistic Signature Scheme*) [11, 35] pour la signature.

Alors que dans le modèle standard, on ne peut être sûr de l'existence d'une transformation des schémas $\mathsf{IND}-\mathsf{CPA}$ sûrs en schémas $\mathsf{IND}-\mathsf{CCA2}$ sûrs, une telle transformation existe dans le modèle de l'oracle aléatoire [51]. De plus, dans [100], Pointcheval a montré une méthode générique très efficace pour construire des schémas $\mathsf{IND}-\mathsf{CCA2}$ sûrs à partir des schémas $\mathsf{OW}-\mathsf{CPA}$ sûrs (le même résultat avec une construction moins efficace a également été indépendamment proposé par Fujisaki et Okamoto [52, 53]). Cela explique la simplicité, et donc, l'efficacité, des schémas prouvés sûrs dans le modèle de l'oracle aléatoire.

1.4 Formalisation de quelques notions de base

Dans les analyses ultérieures, nous utilisons souvent les notions de fonctions négligeables, de famille de fonctions à sens-unique et de famille de fonctions pseudo-aléatoires

[2] Un *prédicat à trappe* (« trapdoor predicate » en anglais) est, informellement, une fonction probabiliste vers {0,1} qui ne peut être efficacement évaluée qu'avec une *trappe*. Goldwasser et Micali [59] ont montré que les prédicats à trappe sont équivalents aux schémas de chiffrement d'un bit $\mathsf{IND}-\mathsf{CPA}$ sûrs qui sont, à leur tour, équivalents aux schémas $\mathsf{IND}-\mathsf{CPA}$ sûrs par une transformation « bit par bit ». Comme le chiffrement d'un bit $\mathsf{IND}-\mathsf{CPA}$ sûr est équivalent au chiffrement d'un bit $\mathsf{OW}-\mathsf{CPA}$ sûr, on déduit que les prédicats à trappe sont équivalents aux schémas $\mathsf{OW}-\mathsf{CPA}$ sûrs.

pour le chiffrement asymétrique et le chiffrement symétrique. Nous présentons ici la définition formelle de ces notions de base :

Définition 1 (Fonction négligeable) *Une fonction $\mu : \mathbb{N} \to \mathbb{R}^+$ est dite négligeable si, pour tout polynôme $p(.)$, il existe un nombre entier N tel que $\mu(n) < \frac{1}{p(n)}$ pour tout $n \geq N$.*

Définition 2 (Famille de fonctions à sens-unique) *Étant donné une famille de fonctions $(f_i : D_i \to \{0,1\}^\star)_{i \in I}$ et deux fonctions $t : \mathbb{N} \to \mathbb{N}$ et $\varepsilon : \mathbb{N} \to \mathbb{R}^+$. On dit que la famille de fonctions $(f_i)_{i \in I}$ est (t, ε)-OW si les conditions suivantes sont satisfaites :*

Facile à échantillonner : *Il existe une machine de Turing polynomiale probabiliste Id et un polynôme $p(.)$ tels que, à partir d'une entrée 1^n, Id retourne un indice $i \in I \cap \{0,1\}^{p(n)}$. Il existe un algorithme Dom qui, à partir de l'entrée $i \in I$, retourne $x \in D_i$.*

Facile à calculer : *Il existe une machine de Turing polynomiale déterministe $Eval$ qui calcule, à partir des entrées $i \in I$ et $x \in D_i$, $f_i(x)$,* i.e. *$Eval(i, x) = f_i(x)$.*

Difficile à inverser : *Pour tout attaquant $\mathcal{A}$, on définit la probabilité de succès de $\mathcal{A}$ d'inverser la famille $(f_i)_{i \in I}$ par :*

$$\mathsf{Succ}^{\mathsf{ow}}_f(\mathcal{A}, n) \stackrel{\mathsf{def}}{=} \Pr_{i \leftarrow Id(1^n), x \leftarrow Dom(i)}[f(\mathcal{A}(i, f(x))) = f(x)].$$

On note que dans cette définition, la probabilité est prise sur i, x et sur tous les aléas dans l'algorithme $\mathcal{A}$. On définit aussi le succès maximal de tous les attaquants qui s'exécutent en un temps borné par $t(n)$ par $\mathsf{Succ}^{\mathsf{ow}}_f(t)$-une fonction de n.
Alors, pour tout n suffisamment grand, $\mathsf{Succ}^{\mathsf{ow}}_f(t)$ est borné par $\varepsilon(n)$.

$(f_i)_{i \in I}$ est également dite famille de fonctions à sens-unique si, pour tout polynôme $t(n)$, il existe une fonction négligeable $\varepsilon(n)$ telle que $(f_i)_{i \in I}$ est (t, ε)-OW.

Considérons l'exemple typique d'un candidat pour des fonctions à sens unique : les fonctions RSA. La famille de fonctions RSA $(RSA_i : D_i \to D_i)_{i \in I}$ est définie par :

- L'ensemble d'indices est composé de couples (N, e), où N est le produit de deux nombres premiers P, Q de taille n et e est un entier plus petit que N et premier avec $(P-1)(Q-1)$.
- Pour chaque $i = (N, e)$, D_i est défini par $D_i \stackrel{\mathsf{def}}{=} \mathbb{Z}^\star_N$.
- Pour tout $x \in \mathbb{Z}^\star_N$, $RSA_{(N,e)}(x) \stackrel{\mathsf{def}}{=} x^e \bmod N$

On remarque que pour tout (N, e), $RSA_{(N,e)}$ est une permutation. La famille de fonctions RSA est, en fait, une famille de permutations. On montre que cette famille satisfait les deux premières conditions d'une famille de fonctions à sens-unique (*i.e.* facile à échantillonner et facile à calculer) grâce aux trois algorithmes $(Id, Dom, Eval)$ suivant :

- L'algorithme Id, sur l'entrée 1^n, choisit uniformément deux nombres premiers P, Q de taille n (pour choisir uniformément un nombre premier de taille n, on prend aléatoirement un nombre entier de taille n et on teste s'il est premier ce on se fait en temps polynomial [1]). Id retourne $N = P.Q$ et un entier e, plus petit que N et premier avec $(P-1)(Q-1)$.

– L'algorithme Dom, sur l'entrée (N, e), choisit aléatoirement $x \in \mathbb{Z}_N^\star$.

– L'algorithme $Eval$ est défini par : $Eval((N, e), x) \stackrel{\mathsf{def}}{=} x^e \bmod N$, pour tout $x \in \mathbb{Z}_N^\star$.

En ce qui concerne la troisième condition pour être une famille de fonctions à sens-unique (*i.e.* dificile à inverser), on remarque que la meilleure méthode connu pour inverser la fonction $RSA_{(N,e)}$ est la factorisation de N. En effet, un problème ouvert bien connu est de savoir si la factorisation de N peut être réduite à une inversion de la fonction $RSA_{(N,e)}$. De plus, lemeilleur algorithme connu pour la factorisation n'est pas polynomial.

Définition 3 (Famille de permutations à sens-unique à trappe) *Étant donné une famille de permutations $(f_i : D_i \to D_i)_{i \in I}$ et deux fonctions $t : \mathbb{N} \to \mathbb{N}$ et $\varepsilon : \mathbb{N} \to \mathbb{R}^+$. On dit que la famille de permutations $(f_i)_{i \in I}$ est (t, ε)-OW à trappe si les conditions suivantes sont satisfaites :*

Facile à échantillonner : *Il existe une machine de Turing polynomiale probabiliste Id et deux polynômes $p(.), q(.)$ tels que, à partir d'une entrée 1^n, Id retourne un indice $i \in I \cap \{0,1\}^{p(n)}$ et une trappe $\mathsf{td} \in \{0,1\}^{q(n)}$. Il existe un algorithme Dom qui, sur l'entrée $i \in I$, retourne $x \in D_i$.*

Facile à calculer : *Il existe une machine de Turing polynomiale déterministe $Eval$ qui calcule, à partir des entrées $i \in I$ et $x \in D_i$, $f_i(x)$,* i.e. *$Eval(I, x) = f_i(x)$.*

Facile à inverser avec trappe : *il existe une machine de Turing polynomiale déterministe $Eval^{-1}$ qui calcule, pour tout $(i, \mathsf{td}) \leftarrow I$ et pour tout $x \in D_i$, $f_i^{-1}(f_i(x))$,* i.e. *$Eval^{-1}(\mathsf{td}, f_i(x)) = x$.*

Difficile à inverser sans trappe : *Pour tout attaquant $\mathcal{A}$, on définit la probabilité de succès de $\mathcal{A}$ d'inverser la famille $(f_i)_{i \in I}$ par :*

$$\mathsf{Succ}_f^{\mathsf{ow}}(\mathcal{A}, n) \stackrel{\mathsf{def}}{=} \Pr_{i \leftarrow Id(1^n), x \leftarrow Dom(i)}[f(\mathcal{A}(i, f(x))) = f(x)].$$

On note que dans cette définition, la probabilité est prise sur i, x et sur tous les aléas dans l'algorithme $\mathcal{A}$. On définit aussi le succès maximal de tous les attaquants qui s'exécutent en un temps borné par $t(n)$ par : $\mathsf{Succ}_f^{\mathsf{ow}}(t)$-une fonction de n.
Alors, pour tout n suffisamment grand, $\mathsf{Succ}_f^{\mathsf{ow}}(t)$ est bornée par $\varepsilon(n)$.

On dit aussi que $(f_i)_{i \in I}$ est une famille de permutations à sens-unique à trappe si, pour tout polynôme $t(n)$, il existe une fonction négligeable $\varepsilon(n)$ telle que $(f_i)_{i \in I}$ soit (t, ε)-OW à trappe.

On voit que la famille de permutations RSA présentée ci-dessus peut être modifiée pour avoir la propriété de trappe : l'algorithme Id est simplement modifié pour retourner l'indice (N, e) et la trappe (N, d) où $d = e^{-1} \bmod (P-1)(Q-1)$. L'algorithme d'inversion $Eval^{-1}$ est défini par $Eval^{-1}((N, d), y) = y^d \bmod N$.

Définition 4 (Famille de fonctions pseudo-aléatoires) *Soit une famille de fonctions $F : Keys(F) \times D \to R$.*
On appelle $U^{D \to R}$ l'ensemble de toutes les fonctions de $D \to R$.
Considérons un attaquant $\mathcal{A}$ qui a accès à un oracle $\mathcal{O}_b$, où :

- $\mathcal{O}_0$ *est une fonction* g *aléatoirement choisie dans* $U^{D\to R}$ *:* $g \xleftarrow{R} U^{D\to R}$*, on dit que* $\mathcal{O}_0$ *est une fonction véritablement aléatoire.*
- $\mathcal{O}_1$ *est une fonction* g *aléatoirement choisie dans la famille* F *:* $g \xleftarrow{R} F$*. Autrement dit,* $k \xleftarrow{R} Keys(F)$ *et* $g \stackrel{\mathsf{def}}{=} F_k$*.*

L'avantage de l'attaquant $\mathcal{A}$ *pour deviner le bit* b *est défini par :*

$$\mathsf{Adv}_F^{\mathsf{prf}}(\mathcal{A}) = 2 \times \Pr_{b,g\xleftarrow{R}U^{D\to R},k\xleftarrow{R}Keys(F)}[\mathcal{O}_0 = g, \mathcal{O}_1 = F_k, \mathcal{A}^{\mathcal{O}_b} = b] - 1.$$

On définit aussi $\mathsf{Adv}_F^{\mathsf{prf}}(t,n)$*, l'avantage maximal de tous les attaquants qui s'exécutent en un temps borné par* t *et qui font au plus* n *requêtes à l'oracle* $\mathcal{O}_b$*.*
On dit que la famille F *est* (ε,t,n)-PRF *si* $\mathsf{Adv}_F^{\mathsf{prf}}(t,n)$ *est borné par* ε*.*

Discussion sur le temps de fonctionnement d'une attaque Dans les définitions ci-dessus et dans celles qui suivent dans cette thèse, le temps d'exécution d'une attaque englobe le temps de calcul et le temps de chargement (ou taille) du code de l'algorithme d'attaque. Ainsi, si les attaques considérées fonctionnent en temps polynomial, le code de l'algorithme a nécessairement une taille polynomiale.

Notons que sans cette convention, certaines hypothèses deviendraient inatteignables. Considérons par exemple un schéma de chiffrement par blocs E_k. Si on ne tient pas compte de la taille du code, le code peut contenir toutes les valeurs de $(E_k(m), m, k)$, pour toutes les clés et tous les clairs. Lors de l'attaque, sur un couple texte clair - texte chiffré, une consultation de la table permet de retrouver la clé k. La résistance aux attaques à clairs-chiffrés connus serait impossible.

2

Analyse des notions de sécurité pour le chiffrement asymétrique

Sommaire

2.1 Notions de sécurité pour le chiffrement asymétrique . . . 18
2.1.1 Schéma de chiffrement asymétrique 19
2.1.2 Niveau de sécurité 19
2.1.3 Puissance de l'attaquant 20
2.2 Relations concrètes entre les notions de sécurité 21
2.2.1 Discussion sur la non-malléabilité 24
2.2.2 Relations entre les notions de sécurité revisitées 26
2.2.3 Nouveau modèle d'attaque : CCAO2 26

Dans les applications réelles, le contexte est étudié afin de choisir le niveau de sécurité adéquat. Puisqu'il n'est pas toujours nécessaire d'utiliser le schéma le plus sûr, il est important d'étudier non seulement les méthodes de construction des schémas mais aussi la relation entre les différentes notions de sécurité. À titre d'exemple, lorsqu'on utilise un schéma de chiffrement asymétrique sémantiquement sûr contre les attaques à chiffrés choisis non-adaptatives, est-il nécessairement non-malléable en considérant seulement des attaques passives ? La réponse est négative [7]. Dans le contexte où la non-malléabilité est importante, un tel schéma (qui est apparemment très sûr) ne suffit pas. La recherche sur les relations entre les notions de sécurité devient très importante : un schéma sûr dans un sens est-il sûr dans un autre sens ?

Les relations entre les notions de sécurité ont été étudiées de manière approfondie par Bellare, Desai, Pointcheval et Rogaway [7], puis par Bellare et Sahai [12]. Dans [7], les auteurs ont montré les séparations et les implications entre les principales notions de sécurité dans le cadre du chiffrement asymétrique. Le résultat est résumé dans la figure 2.1

Dans [12], Bellare et Sahai ont présenté la notion d'attaques parallèles (dénotées PA,

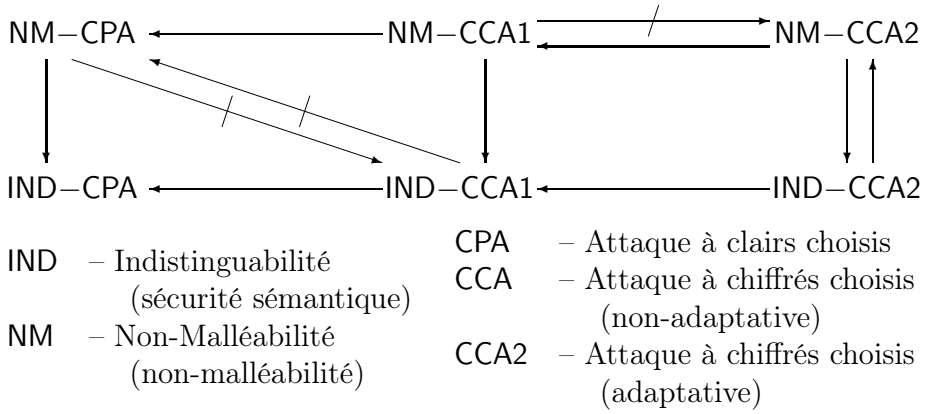

FIG. 2.1 – Relations entre les notions de sécurité

par *parallel attack* en anglais). Dans ce modèle, l'attaquant peut soumettre à la fin (avant de retourner la réponse) un vecteur de chiffrés à l'oracle de déchiffrement. Les auteurs ont montré que la notion NM selon des attaques normales est équivalente à la notion IND selon les attaque parallèles.

Dans notre travail [99], nous considérons les cas concrets en introduisant des niveaux de sécurité $(i, j) - \mathsf{IND}$, $(i, j) - \mathsf{NM}$ où l'attaquant peut poser au maximum i requêtes avant et j requêtes après la réception du challenge. La raison d'une telle notation, plus précise que $\mathsf{IND}-\mathsf{CCA1}$ capturée par $(poly(.), 0)-\mathsf{IND}$ (où $poly(.)$ désigne un polynôme)et $\mathsf{IND}-\mathsf{CCA2}$ capturée par $(poly(.), poly(.))-\mathsf{IND}$, est que l'on peut prouver l'importance de chaque requête : une requête avant de recevoir le challenge ne peut pas être remplacée par une, voire plusieurs requêtes, après avoir reçu le challenge et inversement. Cela contredit l'intuition qu'une requête après réception du challenge est plus importante qu'une requête avant la réception du challenge. Cette notation précise nous permet aussi de mieux appréhender la relation entre l'indistinguabilité et la non-malléabilité.

Comme application, nous introduisons le nouveau modèle d'attaque « à chiffrés choisis post-challenge » (dénoté CCAO2) où l'attaquant ne peut poser des requêtes de déchiffrement qu'après avoir reçu le challenge. Nous montrons que cette notion est indépendante des autres modèles : CCA1 et CCAO2 sont indépendantes et $\mathsf{CCA1} + \mathsf{CCAO2}$ n'implique pas CCA2. D'un point de vue pratique, ce genre d'attaques modélise un scénario très réaliste où l'attaquant commence son attaque à partir du moment où il est au courant de l'importance d'un chiffré spécifique (après que ce dernier a été fabriqué).

2.1 Notions de sécurité pour le chiffrement asymétrique

On rappelle d'abord la notion formelle de schéma de chiffrement asymétrique :

2.1.1 Schéma de chiffrement asymétrique

Un schéma de chiffrement asymétrique π est défini par les trois algorithmes suivants :
- L'*algorithme de génération de clef* $\mathcal{K}$. Sur l'entrée 1^k, où k est le paramètre de sécurité, l'algorithme $\mathcal{K}$ produit un couple $(\mathsf{pk}, \mathsf{sk})$ constitué d'une clef publique et d'une clef secrète. On dénote $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^k)$
- L'*algorithme de chiffrement* $\mathcal{E}$. Étant donné un message m (dans l'espace des textes clairs $\mathcal{M}$) et une clef publique pk, $\mathcal{E}_{\mathsf{pk}}(m)$ produit un texte chiffré c (dans l'espace des chiffrés $\mathcal{C}$) de m. Cet algorithme peut être probabiliste et nécessite alors un aléa supplémentaire $r \in \mathcal{R}$; il est dénoté par $\mathcal{E}_{\mathsf{pk}}(m; r)$.
- L'*algorithme de déchiffrement* $\mathcal{D}$. Étant donné un texte chiffré $c \in \mathcal{C}$ et une clef secrète sk, $\mathcal{D}_{\mathsf{sk}}(c)$ retourne un texte clair $m \in \mathcal{M}$ ou un symbole invalide $\perp$. C'est typiquement un algorithme déterministe.

La condition de consistance requise est que le déchiffrement d'un chiffrement redonne le texte clair original : pour tout couple $(\mathsf{pk}, \mathsf{sk})$ généré par l'algorithme de génération de clef, $\mathcal{D}_{\mathsf{sk}}(\mathcal{E}_{\mathsf{pk}}(m; r)) = m$ pour tout $m \in \mathcal{M}$ et tout $r \in \mathcal{R}$.

Nous rappelons ensuite les formalisations de différentes notions de sécurité qui ont été informellement présentées dans l'introduction.

2.1.2 Niveau de sécurité

Définition 5 (Sens-unique — One-wayness) *Soit un schéma de chiffrement $\pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. Considérons un attaquant $\mathcal{A}$. On définit le succès de $\mathcal{A}$ d'inverser le chiffrement du schéma π par :*

$$\mathsf{Succ}^{\mathsf{ow}}_{\pi}(\mathcal{A}) = \Pr_{m,r}\left[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^k) : \mathcal{A}(\mathsf{pk}, \mathcal{E}_{\mathsf{pk}}(m, r)) = m\right].$$

On définit aussi le succès maximal sur tous les attaquants $\mathcal{A}$, qui fonctionnent en temps borné par t, par $\mathsf{Succ}^{\mathsf{ow}}_{\pi}(t)$. Alors, on dit que π est $(t, \varepsilon)-\mathsf{OW}$ sûr si $\mathsf{Succ}^{\mathsf{ow}}_{\pi}(t)$ est plus petit que ε.

Définition 6 (Indistinguabilité) *Soit un schéma de chiffrement $\pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. Considérons une attaque à deux étapes $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. On définit l'avantage de $\mathcal{A}$ contre le schéma $\pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ au sens de l'indistinguabilité par :*

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{ind}}_{\pi}(\mathcal{A}) &= \left| 2 \times \Pr_{b,r}\left[\begin{array}{l}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^k), (m_0, m_1, s) \leftarrow A_1(\mathsf{pk}), \\ c = \mathcal{E}_{\mathsf{pk}}(m_b, r), b' = A_2(m_0, m_1, s, c) : b' = b\end{array}\right] - 1 \right| \\
&= \left| \Pr_{r}[b' = 1 \,|\, b = 1] - \Pr_{r}[b' = 1 \,|\, b = 0] \right|.
\end{aligned}$$

On insiste sur le fait que $\mathcal{A}_1$ retourne deux messages m_0 et m_1 de même longueur $|m_0| = |m_1|$. On définit aussi l'avantage maximal sur tous les attaquants $\mathcal{A}$, qui fonctionnent en temps borné par t, par $\mathsf{Adv}^{\mathsf{ind}}_{\pi}(t)$. Alors, on dit que π est $(t, \varepsilon)-\mathsf{IND}$ sûr si $\mathsf{Adv}^{\mathsf{ind}}_{\pi}(t)$ est plus petit que ε.

Définition 7 (Non-malléabilité) *Soit un schéma de chiffrement $\pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. Considérons une attaque à deux étapes $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. On définit l'avantage de $\mathcal{A}$ contre le schéma $\pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ au sens de la non-malléabilité par :*

$$\mathsf{Adv}^{\mathsf{nm}}_{\pi}(\mathcal{A}) = \left|\mathsf{Succ}^{M}_{\pi}(\mathcal{A}) - \mathsf{Succ}^{\$}_{\pi}(\mathcal{A})\right|, \text{ avec}$$

$$\left.\begin{array}{rcl} \mathsf{Succ}^{M}_{\pi}(\mathcal{A}) & = & \Pr\left[\begin{array}{c} y \notin \mathbf{y} \ \wedge \ \perp \notin \mathbf{x} \\ \wedge \ R(x, \mathbf{x}) \end{array}\right] \\ \mathsf{Succ}^{\$}_{\pi}(\mathcal{A}) & = & \Pr\left[\begin{array}{c} y \notin \mathbf{y} \ \wedge \ \perp \notin \mathbf{x} \\ \wedge \ R(x^\star, \mathbf{x}) \end{array}\right] \end{array}\right\} \begin{array}{c} \text{sur l'espace de probabilités défini par} \\ \\ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^k), (M, s) \leftarrow A_1(\mathsf{pk}), \\ x, x^\star \leftarrow M, y = \mathcal{E}_{\mathsf{pk}}(x, r), \\ (R, \mathbf{y}) \leftarrow A_2(M, s, y), \mathbf{x} = \mathcal{D}_{\mathsf{sk}}(\mathbf{y}). \end{array}$$

On définit aussi l'avantage maximal sur tous les attaquants $\mathcal{A}$, qui fonctionnent en temps borné par t, par $\mathsf{Adv}^{\mathsf{nm}}_{\pi}(t)$. Alors, on dit que π est $(t, \varepsilon)-\mathsf{NM}$ sûr si $\mathsf{Adv}^{\mathsf{nm}}_{\pi}(t)$ est plus petit que ε.

2.1.3 Puissance de l'attaquant

Définition 8 (Attaque à chiffrés choisis non-adaptative) *Un attaquant est appelé* à chiffrés choisis non-adaptatif*,(dénoté* CCA1*) s'il ne peut accéder à l'oracle de déchiffrement après avoir eu connaissance du challenge.*

Définition 9 (Attaque à chiffrés choisis adaptative) *Un attaquant est appelé* à chiffrés choisis adaptatif*, (dénoté* CCA2*) s'il peut accéder à l'oracle de déchiffrement à tout instant,* i.e. *avant et après la réception du challenge ; la seule restriction étant de ne pas faire de requête sur le challenge lui-même.*

Pour plus de généralité, nous présentons une définition plus précise avec un (i, j)-*attaquant* qui peut faire au plus i requêtes (resp. j requêtes) avant la réception du challenge (resp. après la réception du challenge)

Définition 10 (Attaque à chiffrés choisis) *Un attaquant est un* (i, j)-attaquant à chiffrés choisis*, (dénoté $(i, j)-\mathsf{CCA}$) s'il peut faire au maximum i requêtes (resp. j requêtes) à l'oracle avant (resp. après) la réception du challenge ; la seule restriction étant de ne pas faire de requête sur le challenge lui-même.*

Notation. Un schéma de chiffrement $\pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ est dit $(t, \varepsilon)-\mathsf{XXX}-\mathsf{YYY}$ sûr si pour tout YYY-attaquant $\mathcal{A}$ au sens de la sécurité XXX en temps t, l'avantage de $\mathcal{A}$ est borné par ε. Ici, XXX peut être IND ou NM, et YYY peut être CPA, $\mathsf{CCA1}$, $\mathsf{CCA2}$, ou $(i, j)-\mathsf{CCA}$. Pour simplifier la notation, on dit aussi que π est $(t, \varepsilon, i, j)-\mathsf{IND}$ sûr (resp. $(t, \varepsilon, i, j)-\mathsf{NM}$ sûr) si pour tout $(i, j)-\mathsf{CCA}$ attaquant $\mathcal{A}$ dont le temps de calcul est borné par t, $\mathsf{Adv}^{\mathsf{ind}}_{\pi}(\mathcal{A}) \leq \varepsilon$ (resp. $\mathsf{Adv}^{\mathsf{nm}}_{\pi}(\mathcal{A}) \leq \varepsilon$.)

2.2 Relations concrètes entre les notions de sécurité

Dans cette partie, les relations concrètes entre les notions de sécurité seront démontrées. On verra ensuite que les preuves des résultats précédents [7] peuvent être déduites de notre théorème 15.

Nous allons montrer quelques différences non-intuitives dans les classes $(i, j)-\mathsf{IND}$: une requête de déchiffrement dans la première étape (avant la connaissance du challenge) ne peut pas être retardée à la deuxième étape (après connaissance du challenge) et inversement. De manière formelle, nous prouverons qu'une requête de plus dans la première étape peut donner plus d'avantage à l'attaquant que plusieurs requêtes dans la deuxième étape. L'inverse est aussi vrai, *i.e.* une requête dans la deuxième étape ne peut pas être remplacée par plusieurs requêtes dans la première étape.

Pour cela, nous introduisons un nouveau problème calculatoire lié aux fonctions pseudo-aléatoires.

Définition 11 *Étant donné une fonction G, aléatoirement choisie dans une famille $\mathcal{G}$ de fonctions/permutations de E dans E, et une attaque à deux étapes $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ où $\mathcal{A}_1$ et $\mathcal{A}_2$ ne peuvent respectivement faire qu'au plus m et $n-1$ requêtes à G. Nous définissons $\mathsf{Succ}^{m,n}_{\mathcal{G}}(\mathcal{A})$ la probabilité pour $\mathcal{A}$ de calculer $G^n(v)$ sur une instance aléatoire $v \in E$:*

$$\mathsf{Succ}^{m,n}_{\mathcal{G}}(\mathcal{A}) = \Pr[G \xleftarrow{R} \mathcal{G}; v \xleftarrow{R} E; s \leftarrow \mathcal{A}_1^G : \mathcal{A}_2^G(v, s) = G^n(v)].$$

Nous dénotons aussi $\mathsf{Succ}^{m,n}_{\mathcal{G}}(t)$ la valeur maximale de $\mathsf{Succ}^{m,n}_{\mathcal{G}}(\mathcal{A})$ sur tous les attaquants dont le temps de calcul est borné par t.

Proposition 12 *Soit $\mathcal{G}$ une famille de fonctions/permutations pseudo-aléatoires de E dans E. Pour tout ℓ tel que le cardinal de E soit supérieur à 2^ℓ, on a :*

$$\mathsf{Succ}^{m,n}_{\mathcal{G}}(t) \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathcal{G}}(m + 2n - 1, t) + \frac{mn+1}{2^\ell}.$$

Preuve. Considérons un attaquant $\mathcal{A}$ contre la famille $\mathcal{G}$ dont le succès $\mathsf{Succ}^{m,n}_{\mathcal{G}}(t)$ est non-négligeable. Nous construisons un attaquant-PRF $\mathcal{B}$ tel que $\mathsf{Succ}^{m,n}_{\mathcal{G}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathcal{G}}(\mathcal{B})$. L'attaquant $\mathcal{B}$ fonctionne de la manière suivante : quand $\mathcal{A}$ fait une requête à G, $\mathcal{B}$ fait la même requête à $\mathcal{O}_b$ et transmet la réponse à $\mathcal{A}$ (au plus $m + n - 1$ requêtes au total.) À la fin du processus, $\mathcal{A}$ retourne x. Alors, $\mathcal{B}$ fait successivement des requêtes à l'oracle $\mathcal{O}_b$ pour obtenir $y = \mathcal{O}_b^n(v)$. Si $x = y$, $\mathcal{B}$ retourne le bit $b' = 1$, sinon $\mathcal{B}$ retourne le bit $b' = 0$.

– si $b = 1$, $\mathcal{B}$ a accès à G et $y = \mathcal{O}_b^n(v) = G^n(v)$. $\mathcal{B}$ gagne toujours le jeu si $\mathcal{A}$ gagne : $b' = 1$ signifie que $x = y$.

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathcal{G}}(\mathcal{B} \,|\, b = 1) = 2\Pr[x = y \,|\, b = 1] - 1 = 2\mathsf{Succ}^{m,n}_{\mathcal{G}}(\mathcal{A}) - 1.$$

– si $b = 0$, $y = \mathcal{O}_b^n(v)$ est parfaitement aléatoire et indépendant de la vue de $\mathcal{A}$, sauf si $\mathcal{A}_1$ a fait une requête sur $\mathcal{O}_b^i(v)$ (pour $0 \leq i < n$). On obtient donc :

$$\mathsf{Adv}_{\mathcal{G}}^{\mathsf{prp}}(\mathcal{B} \mid b = 0) = 2\Pr[x = y \mid b = 0] - 1 \leq 2 \times \left(\frac{mn}{2^\ell} + \frac{1}{2^\ell}\right) - 1.$$

En combinant les deux cas, en sachant que b est un bit aléatoire, on obtient le résultat. □

La construction et la proposition qui suivent seront utilisées plusieurs fois dans nos preuves de sécurité.

Définition 13 *Soient $\pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ un schéma de chiffrement public et f une permutation sur $\mathcal{M}$, l'inverse de f est dénoté par f^{-1}. Nous définissons un nouveau schéma de chiffrement $\pi^{(f)} = (\mathcal{K}^{(f)}, \mathcal{E}^{(f)}, \mathcal{D}^{(f)})$:*

$$\mathcal{M}^{(f)} = \mathcal{M} \qquad \mathcal{R}^{(f)} = \mathcal{R} \qquad \mathcal{C}^{(f)} = \mathcal{C}$$

Algorithme $\mathcal{K}^{(f)}(1^k)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^k)$
$\mathsf{pk}^{(f)} \leftarrow \mathsf{pk}||f||f^{-1}$
$\mathsf{sk}^{(f)} \leftarrow \mathsf{sk}$
retourner $(\mathsf{pk}^{(f)}, \mathsf{sk}^{(f)})$

Algorithme $\mathcal{E}^{(f)}_{\mathsf{pk}^{(f)}}(m, r)$
$\mathsf{pk}||f||f^{-1} \stackrel{\mathsf{def}}{=} \mathsf{pk}^{(f)}$
retourner $\mathcal{E}_{\mathsf{pk}}(f(m), r)$

Algorithme $\mathcal{D}^{(f)}_{\mathsf{sk}^{(f)}}(c)$
$\mathsf{sk} \stackrel{\mathsf{def}}{=} \mathsf{sk}^{(f)}$
retourner $f^{-1}(\mathcal{D}_{\mathsf{sk}}(c))$

Proposition 14 *Soient $\pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ un schéma de chiffrement et f une permutation, où f et f^{-1} sont efficacement calculables, π et $\pi^{(f)}$ ont le même niveau de sécurité au sens de l'indistinguabilité quel que soit le modèle d'attaque :*

$$\mathsf{Adv}_{\pi}^{\mathsf{ind-yyy}}(t) \leq \mathsf{Adv}_{\pi^{(f)}}^{\mathsf{ind-yyy}}(t + 2T_f + q_d T_{f^{-1}}) \leq \mathsf{Adv}_{\pi}^{\mathsf{ind-yyy}}(t + (2 + q_d)(T_f + T_{f^{-1}})),$$

où T_f ($T_{f^{-1}}$ resp.) est la borne supérieure du le temps nécessaire pour évaluer f (f^{-1} resp.)

Preuve. Nous montrons d'abord que si $\pi^{(f)}$ est sûr, alors π l'est aussi. Rappelons que f et f^{-1} sont efficacement calculables et inclus dans la clef publique. Considérons un attaquant $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ contre π, nous construisons un attaquant $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ contre $\pi^{(f)}$: à chaque fois que $\mathcal{A}$ fait une requête de déchiffrement c à l'oracle $\mathcal{D}_{\mathsf{sk}}$, $\mathcal{B}$ fait la même requête à l'oracle de déchiffrement $\mathcal{D}^{(f)}_{\mathsf{sk}^{(f)}}$. $\mathcal{B}$ reçoit la réponse m et transmet $f(m)$ à $\mathcal{A}$. Quand $\mathcal{A}_1$ retourne deux candidats m_0 et m_1, $\mathcal{B}_1$ calcule $f^{-1}(m_0)$ et $f^{-1}(m_1)$. Finalement, quand $\mathcal{A}$ devine b', $\mathcal{B}$ retourne aussi cette valeur. Il est clair que l'avantage de $\mathcal{B}$ est le même que celui de $\mathcal{A}$, mais il a besoin d'évaluations additionnelles : deux pour calculer f et q_d pour calculer f^{-1}, où q_d est le nombre de requêtes à l'oracle de déchiffrement :

$$\mathsf{Adv}_{\pi}^{\mathsf{ind-yyy}}(t) \leq \mathsf{Adv}_{\pi^{(f)}}^{\mathsf{ind-yyy}}(t + 2T_f + q_d T_{f^{-1}}).$$

Puisque $\pi = \pi^{(f)(f^{-1})}$, et que f et f^{-1} sont publiques et efficacement calculables, on conclut facilement. □

Théorème 15 *Sous l'hypothèse de l'existence de permutations à sens-unique à trappe, pour tout couple* (m, n)*, il existe un schéma de chiffrement qui est* $(m, N)-\mathsf{IND}$ *et* $(M, n)-\mathsf{IND}$ *sûr mais pas* $(m+1, n+1)-\mathsf{IND}$ *sûr quels que soient* M *et* N*.*

Idée de la preuve. L'hypothèse de l'existence de permutations à sens-unique à trappe est équivalente à l'hypothèse de l'existence de schémas de chiffrement $\mathsf{IND}-\mathsf{CCA2}$-sûrs [43, 44]. Alors, on suppose qu'il existe un schéma $\pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ qui soit $\mathsf{IND}-\mathsf{CCA2}$ sûr. Il est, par conséquent, $(i, j)-\mathsf{IND}$ sûr pour tout (i, j). Supposons aussi que f soit une permutation à sens unique à trappe vers $\mathcal{M}$. D'après la proposition 14, le schéma $\pi^{(f)}$ est aussi $\mathsf{IND}-\mathsf{CCA2}$ sûr quand la trappe pour calculer f^{-1} est incluse dans la clef publique. Nous transformons $\pi^{(f)}$ en un nouveau schéma de chiffrement $\pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ qui n'est plus $(m+1, n+1)-\mathsf{IND}$ sûr, mais reste à la fois $(m, N)-\mathsf{IND}$ sûr et $(M, n)-\mathsf{IND}$ sûr. Notons que dans π', la trappe pour calculer f^{-1} est désormais incluse dans la clef secrète. Ce schéma fonctionne comme suit :

- On dénote I_M un élément spécifique de $\mathcal{M}$ et pose $p_M = f^{-1}(I_M)$.
- On sélectionne deux familles : l'une de fonctions pseudo-aléatoires $\mathcal{F} = \{F_K : K \in \{0,1\}^k\}$ de $\mathcal{C}$ vers $\mathcal{C}$ et l'autre de permutations pseudo-aléatoires $\mathcal{G} = \{G_K : K \in \{0,1\}^k\}$ de $\mathcal{C}$ vers $\mathcal{C}$. On suppose, de plus, que le cardinal de $\mathcal{C}$ est plus grand que 2^ℓ.

Algorithme $\mathcal{K}'(1^k)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^k)$
$I_M \overset{R}{\leftarrow} \mathcal{M}, K_f, K_g \overset{R}{\leftarrow} \{0,1\}^k$
$\mathsf{pk}' \leftarrow \mathsf{pk}||f||I_M||m||n$
$\mathsf{sk}' \leftarrow \mathsf{sk}||f^{-1}||K_f||K_g$
retourner $(\mathsf{pk}', \mathsf{sk}')$

Algorithme $\mathcal{E}'_{\mathsf{pk}'}(\mu, r)$
$\mathsf{pk}||f||I_M||m||n \overset{\mathsf{def}}{=} \mathsf{pk}'$
$\varphi \leftarrow f(\mu)$
retourner $0||\mathcal{E}_{\mathsf{pk}}(\varphi, r)||\epsilon$

Algorithme $\mathcal{D}'_{\mathsf{sk}'}(b||c||z)$
$\mathsf{sk}||f^{-1}||K_f||K_g \overset{\mathsf{def}}{=} \mathsf{sk}'$
1. si $(b = 0 \wedge z = \epsilon)$ retourner $f^{-1}(\mathcal{D}_{\mathsf{sk}}(c))$
2. si $(b = 1 \wedge z = \epsilon)$ retourner $F_{K_f}(c)$
3. si $(b = 2 \wedge z = \epsilon)$ retourner $G_{K_g}(c)$
4. si $(b = 1 \wedge z = F^n_{K_f}(c) \wedge \mathcal{D}(c) = G^m_{K_g}(I_M))$ retourner $f^{-1}(G^m_{K_g}(I_M))$

sinon, retourner $\perp$

La construction est fondée sur l'idée que $m+1$ requêtes de déchiffrement dans la première étape permettent de déterminer un texte clair spécifique $\mu = f^{-1}(G^m_{K_g}(I_M))$. La spécificité vient du fait que, en faisant $n+1$ requêtes de déchiffrement dans la deuxième étape, il est possible de vérifier si un texte chiffré $0\|c\|\epsilon$ est un chiffrement de μ : d'abord on calcule $F^n_{K_f}(c)$ par n requêtes de déchiffrement, puis soumet $1\|c\|F^n_{K_f}(c)$ à l'oracle de

déchiffrement. Alors, on peut facilement construire un $(n+1, m+1)-\mathsf{CCA}$ attaquant qui casse le schéma. De plus, avec une preuve similaire à celle de la proposition 14, on peut montrer que le schéma est sûr face aux $(n, M)-\mathsf{CCA}$ attaquants et $(N, m)-\mathsf{CCA}$ attaquants.

2.2.1 Discussion sur la non-malléabilité

Nous discutons brièvement la notion « générale » de non-malléabilité (dénotée $\mathsf{CNM}^{(k)}$) dans laquelle l'attaquant produit, à la fin, un vecteur de chiffrés de dimension k, au lieu d'un seul chiffré. Dans [12], Bellare et Sahai ont présenté la notion d'attaques parallèles, dénotées PA (plus précisément $\mathsf{PA}^{(k)}$ dans notre contexte). Dans les $\mathsf{PA}^{(k)}$, après la dernière requête à l'oracle de déchiffrement, l'attaquant peut faire une requête sur un vecteur de chiffrés de taille k. Les $\mathsf{PA}^{(k)}$ peuvent être divisées en trois catégories : $\mathsf{PA0}$, $\mathsf{PA1}$ et $\mathsf{PA2}$, selon la possibilité d'accès à l'oracle de déchiffrement et sans tenir compte du vecteur de chiffrés supplémentaire (jamais, avant, avant et après la réception du challenge). Bellare et Sahai ont montré que $\mathsf{IND}-\mathsf{PAX}$ est équivalente à $\mathsf{CNM}^{(k)}-\mathsf{CCAX}$, où $\mathsf{CCA0}$ est en fait CPA. Leur résultat peut être traduit dans notre formalisation par le théorème suivant.

Théorème 16 *Les notions $(m, n)-\mathsf{IND}-\mathsf{PA}^{(k)}$ et $(m, n)-\mathsf{CNM}^{(k)}$ sont équivalentes. En d'autres termes, quel que soit le schéma de chiffrement $\pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$:*

$$\frac{1}{2} \times \mathsf{Adv}_{\pi}^{(m,n)-\mathsf{ind}-\mathsf{pa}^{(k)}}(t) \leq \mathsf{Adv}_{\pi}^{(m,n)-\mathsf{cnm}^{(k)}}(t) \leq \mathsf{Adv}_{\pi}^{(m,n)-\mathsf{ind}-\mathsf{pa}^{(k)}}(t + T_R),$$

où T_R est une borne supérieure sur temps de calcul de la relation R.

Preuve. Considérons un $(m, n)-\mathsf{IND}-\mathsf{PA}^{(k)}$-attaquant $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}'_2)$ contre π. Il s'agit d'une attaque-IND classique à deux étapes. La deuxième étape est décomposée en deux parties : $\mathcal{A}_2$ a accès à l'oracle de déchiffrement et retourne un vecteur de chiffrés ; $\mathcal{A}'_2$ reçoit le vecteur de textes clairs correspondants et retourne le bit d'estimation sans avoir besoin de l'accès à l'oracle de déchiffrement. Nous construisons un $(m, n)-\mathsf{CNM}^{(k)}$-attaquant $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ de la manière suivante :

- $\mathcal{B}_1$ exécute $\mathcal{A}_1$: à chaque fois que $\mathcal{A}_1$ fait une requête de déchiffrement, $\mathcal{B}_1$ fait la même requête à l'oracle de déchiffrement, puis transmet la réponse à $\mathcal{A}_1$. Quand $\mathcal{A}_1$ retourne deux candidats (m_0, m_1) et s, on définit et retourne une distribution uniforme $\mathcal{M} = \{m_0, m_1\}$ et l'information s ;
- Le challengeur choisit aléatoirement m suivant la distribution $\mathcal{M}$. De manière équivalente, il s'agit de choisir $b \xleftarrow{R} \{0, 1\}$ et poser $m = m_b$. Ensuite, on fabrique le challenge $c = \mathcal{E}_{\mathsf{pk}}(m, r)$ où $r \xleftarrow{R} R$;
- $\mathcal{B}_2$ reçoit l'information s et le challenge c. Il les transmet à $\mathcal{A}_2$. $\mathcal{B}_2$ exécute $\mathcal{A}_2$: à chaque fois que $\mathcal{A}_2$ fait une requête de déchiffrement, $\mathcal{B}_2$ fait la même requête à l'oracle de déchiffrement, puis transmet la réponse à $\mathcal{A}_2$. Quand $\mathcal{A}_2$ retourne le vecteur de textes chiffrés $\mathbf{y}$ et l'information s', $\mathcal{B}_2$ retourne $(R, \mathbf{y})$ où la relation R est définie par : $R(\mathbf{x}, m)$ retourne $(m = m_0) \oplus A'_2(\mathbf{x}, s')$.

Par définition, si on considère $\tilde{m}$ est défini selon $\mathcal{M}$, ce qui est équivalent à choisir aléatoirement un bit d indépendant de b et b', puis poser $\tilde{m} = m_d$, alors $\mathsf{Adv}_{\pi}^{(m,n)-\mathsf{ind}-\mathsf{cnm}^{(k)}}(\mathcal{A})$ est égal à :

$$\begin{aligned}
& \Pr[R(\mathbf{x}, m)] - \Pr[R(\mathbf{x}, \tilde{m})] = \Pr[R(\mathbf{x}, m_b)] - \Pr[R(\mathbf{x}, m_d)] \\
= \;& \frac{1}{2} \times \Big(\Pr[R(\mathbf{x}, m_b)] - \Pr[R(\mathbf{x}, m_{\bar{b}})] \Big) \\
= \;& \frac{1}{2} \times \Big(\Pr[(m_0 = m_b) \oplus A'_2(\mathbf{x}, s')] - \Pr[(m_0 = m_{\bar{b}}) \oplus A'_2(\mathbf{x}, s')] \Big) \\
= \;& \frac{1}{2} \times \Big(\Pr[(b = 0) \oplus (b' = 1)] - \Pr[(b = 0) \oplus (b' = 1)] \Big) \\
= \;& \frac{1}{2} \times \Big(\Pr[b = b'] - \Pr[b \neq b'] \Big) = \frac{1}{2} \times \mathsf{Adv}_{\pi}^{(m,n)-\mathsf{ind}-\mathsf{pa}^{(k)}}(\mathcal{B}).
\end{aligned}$$

Remarquons que le temps de calcul de $\mathcal{B}$ est exactement égal à celui de $\mathcal{A}$. On passe maintenant à la relation de droite du théorème. Considérons un $(m,n)-\mathsf{CNM}^{(k)}$-attaquant $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ contre π. Nous construisons un $(m,n)-\mathsf{IND}-\mathsf{PA}^{(k)}$-attaquant $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}'_2)$ de la manière suivante :

- $\mathcal{B}_1$ exécute $\mathcal{A}_1$: à chaque fois que $\mathcal{A}_1$ fait une requête de déchiffrement, $\mathcal{B}_1$ fait la même requête à l'oracle de déchiffrement, puis transmet la réponse à $\mathcal{A}_1$. Quand $\mathcal{A}_1$ retourne une distribution $\mathcal{M}$ et l'information s, on choisit aléatoirement deux textes clairs (m_0, m_1) suivant la distribution $\mathcal{M}$. $\mathcal{B}_1$ retourne ces deux textes ainsi que l'information s ;
- Le challengeur choisit aléatoirement $b \xleftarrow{R} \{0,1\}$ et $r \xleftarrow{R} R$, puis fabrique le challenge $c = \mathcal{E}_{\mathsf{pk}}(m_b, r)$;
- $\mathcal{B}_2$ reçoit l'information s et le challenge c. Il les transmet à $\mathcal{A}_2$. $\mathcal{B}_2$ exécute $\mathcal{A}_2$: à chaque fois que $\mathcal{A}_2$ fait une requête de déchiffrement, $\mathcal{B}_2$ fait la même requête à l'oracle de déchiffrement, puis transmet la réponse à $\mathcal{A}_2$. Quand $\mathcal{A}_2$ retourne $(R, \mathbf{y})$, $\mathcal{B}_2$ retourne $(R, \mathbf{y})$. $\mathcal{B}'_2$ reçoit donc le vecteur de textes clairs correspondant $\mathbf{x}$. $\mathcal{B}'_2$ vérifie si $R(\mathbf{x}, m_0)$ est vraie. Si oui, il retourne $b' = 0$, sinon $b' = 1$.

$$\begin{aligned}
\mathsf{Adv}_{\pi}^{(m,n)-\mathsf{ind}-\mathsf{pa}^{(k)}}(\mathcal{B}) &= \Pr[b' = 0 \,|\, b = 0] - \Pr[b' = 0 \,|\, b = b_1] \\
&= \Pr[R(\mathbf{x}, m_0) \,|\, b = 0] - \Pr[R(\mathbf{x}, m_0) \,|\, b = 1] \\
&= \Pr[R(\mathbf{x}, m_0) \,|\, b = 0] - \Pr[R(\mathbf{x}, m_1) \,|\, b = 0] \\
&= \mathsf{Adv}_{\pi}^{(m,n)-\mathsf{ind}-\mathsf{cnm}^{(k)}}(\mathcal{A}).
\end{aligned}$$

Remarquons que le temps de calcul de $\mathcal{B}$ est égal à celui de $\mathcal{A}$, augmenté du temps d'évaluation de R. □

Remarque. Soient les identifications suivantes,

$$(m,n)-\mathsf{IND}-\mathsf{PA}^{(1)} = (m, n+1)-\mathsf{IND} \qquad (m,n)-\mathsf{CNM}^{(1)} = (m,n)-\mathsf{NM},$$

on a $(m, n+1)-\mathsf{IND} = (m,n)-\mathsf{NM}$.

2.2.2 Relations entre les notions de sécurité revisitées

Dans [7], les auteurs ont montré plusieurs relations entre les notions de sécurité. Leur résultats sont résumés dans la figure 2.1 :

On peut voir que tous ces résultats se déduisent de notre théorème 15. En effet, les implications et séparations non triviales peuvent être montrées de la façon suivante :

1. $\mathsf{IND}-\mathsf{CCA1} \not\rightarrow \mathsf{NM}-\mathsf{CPA}$: il s'agit en effet de
$$(N,0)-\mathsf{IND} \not\rightarrow (0,1)-\mathsf{IND} = (0,0)-\mathsf{NM} = \mathsf{NM}-\mathsf{CPA}.$$
2. $\mathsf{NM}-\mathsf{CPA} \not\rightarrow \mathsf{IND}-\mathsf{CCA1}$: il s'agit en effet de
$$\mathsf{NM}-\mathsf{CPA} = (0,0)-\mathsf{NM} = (0,1)-\mathsf{IND} \not\rightarrow (N,0)-\mathsf{IND}.$$
3. $\mathsf{IND}-\mathsf{CCA2} \rightarrow \mathsf{NM}-\mathsf{CCA2}$: il s'agit en effet de
$$(N,M+1)-\mathsf{IND} = (N,M)-\mathsf{NM}.$$
4. $\mathsf{NM}-\mathsf{CCA1} \not\rightarrow \mathsf{NM}-\mathsf{CCA2}$: il s'agit en effet de
$$(N,0)-\mathsf{NM} = (N,1)-\mathsf{IND} \not\rightarrow (0,2)-\mathsf{IND} = (0,1)-\mathsf{NM} \not\rightarrow (N,M)-\mathsf{NM}.$$

2.2.3 Nouveau modèle d'attaque : CCAO2

En guise d'application, nous introduisons un nouveau modèle d'attaque appelé *attaque à chiffrés choisis post-challenge* et dénoté CCAO2 (pour indiquer des attaques à chiffrés choisis mais seulement dans la deuxième étape, *i.e.* après la réception du challenge.) Ce nouveau scénario complète la figure 2.1 avec la notion de sécurité de $(0, \mathrm{poly}(\cdot))-\mathsf{ind}$. Par ailleurs, d'un point de vue pratique, il modélise des situations réalistes puisque le contrôle dont dispose l'attaquant sur la distribution a priori des clairs est limité. De plus, il couvre également les situations où l'attaquant ne commence l'attaque qu'après avoir pris conscience de l'importance d'un chiffré spécifique.

Grâce au théorème 15, nous montrons que ce modèle d'attaque est indépendant d'autres modèles connus dans la littérature.

Définition 17 (Attaques post-challenge) *Un attaquant est appelé* attaquant à chiffrés choisis post-challenge *(dénoté* CCAO2*-attaquant) s'il ne peut accéder à l'oracle de déchiffrement qu'après que le challenge est connu et qu'il est contraint de ne pas faire de requête sur le challenge à cet oracle.*

Ce nouveau modèle d'attaque, combiné avec les objectifs classiques de sécurité, nous donne les deux notions de sécurité : IND−CCAO2 et NM−CCAO2. Elles sont indépendantes des notions précédentes, à l'exception des implications triviales. D'abord, il est clair que, pour n'importe quel XXX, XXX−CCA2 implique XXX−CCA1 et XXX−CCAO2. Cependant, grâce au résultat ci-dessus, nous prouvons que l'inverse n'est pas vrai. En effet, nous avons les corollaires suivants :

Corollaire 18 IND−CCAO2 *et* IND−CCA1 *sont deux notions indépendantes. En d'autres termes, sous l'hypothèse de l'existence de permutations à sens-unique à trappe, il existe un schéma* IND−CCA1 *sûr mais pas* IND−CCAO2 *sûr. De même, il existe un schéma* IND−CCAO2 *sûr mais pas* IND−CCA1 *sûr.*

Corollaire 19 IND−CCA1 *et* IND−CCAO2 *n'impliquent pas, même pris ensemble,* IND−CCA2*. En d'autres termes, sous l'hypothèse de l'existence de permutations à sens-unique à trappe, il existe un schéma à la fois* IND−CCA1 *sûr et* IND−CCAO2 *sûr mais pas* IND−CCA2 *sûr.*