RÈGLES ET RECOMMANDATIONS CONCERNANT LE CHOIX ET LE DIMENSIONNEMENT DE L'ENSEMBLE DES MÉCANISMES CRYPTOGRAPHIQUES

Annexe à l'Arrêté Ministériel n° 2018-635
du 2 juillet 2018

ANNEXE AU « JOURNAL DE MONACO » N° 8.390

DU 13 JUILLET 2018

TABLE DES MATIÈRES

1 Introduction 2
1.1 Objectif de l'annexe 2
1.2 Limites du champ d'application 2
1.3 Définition des règles et recommandations 3
1.4 Organisation de l'annexe 3
1.5 Mise à jour de l'annexe 3
2 Règles et recommandations 3
2.1 Cryptographie symétrique 3
2.1.1 Taille de clé symétrique 3
2.1.2 Chiffrement symétrique 4
2.1.2.1 Chiffrement par bloc 4
2.1.2.2 Chiffrement par flot 5
2.1.3 Authentification et intégrité de messages 6
2.2 Cryptographie asymétrique 7
2.2.1 Problèmes mathématiques asymétriques 7
2.2.1.1 Factorisation 7
2.2.1.2 Le « Logarithme discret » 7
2.2.2 Chiffrement asymétrique 8
2.2.3 Signature asymétrique 9
2.2.4 Authentification d'entités et établissement de clé 9
2.3 Fonctions de hachage 10
2.4 Génération d'aléa cryptographique 10
2.4.1 Architecture d'un générateur d'aléa 11
2.4.2 Générateur physique d'aléa 12
2.4.3 Retraitement algorithmique 12
2.5 Gestion de clés 12

1 Introduction

1.1 Objectif de l'annexe

La présente annexe traite des règles et recommandations concernant le choix et le dimensionnement de l'ensemble des mécanismes cryptographiques.

Il existe deux autres documents annexés à deux arrêtés ministériels distincts :

- L'arrêté ministériel n° 2018-637 du 2 juillet 2018 portant application de l'article 54 de l'Ordonnance Souveraine n° 3.413 du 29 août 2011 portant diverses mesures relatives à la relation entre l'Administration et l'administré, modifiée concernant la « Gestion des clés cryptographiques - Règles et recommandations concernant la gestion des clés utilisées dans des mécanismes cryptographiques », traitant plus spécifiquement des aspects liés à la création, la distribution et la manipulation de clés ;

- L'arrêté ministériel n° 2018-636 du 2 juillet 2018 portant application de l'article 54 de l'Ordonnance Souveraine n° 3.413 du 29 août 2011 portant diverses mesures relatives à la relation entre l'Administration et l'administré, modifiée concernant l'« Authentification – Règles et recommandations concernant les mécanismes d'authentification » traitant plus spécifiquement des aspects liés à l'utilisation de mots de passe, de cartes mémoire, de clés de déverrouillage pour accéder à un système d'information.

1.2 Limites du champ d'application

Sont explicitement exclus de ce document :

- la recommandation de mécanismes cryptographiques précis permettant d'atteindre les différents niveaux de robustesse cryptographique définis dans ce document, les aspects liés à l'implantation des mécanismes et en particulier au choix du support ainsi qu'à la sécurité de l'implantation face aux attaques par canaux auxiliaires (timing attack, simple power analysis [SPA], differential power analysis [DPA], higher order differential power analysis [HO-DPA], electromagnetic power analysis [EMA]...) ou par injection de faute (differential fault analysis [DFA]) ;

- les méthodes d'évaluation des mécanismes cryptographiques, qui reposent avant tout sur une connaissance précise de l'état de l'art en cryptographie ;

- les méthodes d'analyse de menaces et de développement de produits cryptographiques menant à choisir les mécanismes cryptographiques permettant d'assurer les fonctions de sécurité identifiées ainsi que les niveaux de robustesse cryptographique nécessaires ;

- les liens entre niveau de robustesse d'un mécanisme cryptographique et niveau de robustesse d'un produit tel que défini dans les processus de qualification ou d'évaluation selon une méthode normalisée telle que les Critères Communs ;

- les fournitures nécessaires à l'évaluation de mécanismes cryptographiques.

1.3 Définition des règles et recommandations

Les **règles** définissent des principes qui doivent être suivis par tout mécanisme. L'observation de ces règles est une condition nécessaire à la sécurité du mécanisme. Cependant, le fait de suivre l'ensemble des règles, qui sont par nature très génériques, n'est pas suffisant pour garantir la robustesse du mécanisme cryptographique ; seule une analyse spécifique permet de s'en assurer.

À côté des règles, le présent document définit également des **recommandations**. Elles ont pour but de guider le choix de certaines fonctions mathématiques types primitives et d'inciter à certains dimensionnements permettant un gain considérable en termes de sécurité, pour un coût souvent modique. Il va de soi qu'en tant que recommandations, leur application peut être plus librement modulée en fonction d'autres impératifs tels que des contraintes de performance ou de coût.

L'objectif est de contribuer à une amélioration constante de la qualité des produits de sécurité. À ce titre, le suivi des règles énoncées dans ce document doit être considéré comme une démarche saine permettant de se prémunir contre de nombreuses erreurs de conception ainsi que contre d'éventuelles faiblesses non décelées lors de l'évaluation des mécanismes cryptographiques.

La définition des règles et des recommandations prend également en compte certaines hypothèses classiques telles que la loi de Moore sur l'évolution de la puissance de calcul disponible. Il va cependant de soi qu'une telle analyse ne peut tenir compte d'éventuels événements « catastrophiques » tels qu'une cryptanalyse opérationnelle de l'AES (*Advanced Encryption Standard*) ou la découverte d'une méthode de factorisation efficace sur de grands nombres.

Par ailleurs, l'estimation des niveaux de résistance qui seront nécessaires afin de garantir la sécurité à 10 ou 20 ans des informations est délicate. Elle est cependant requise par de nombreuses applications comme par exemple le maintien de la confidentialité de certaines informations ou la signature électronique qui nécessite souvent une validité à long terme.

De plus, lors de la définition d'un produit, il est nécessaire d'avoir une vision dont le terme est dicté par la durée de vie envisagée. Il est bien entendu possible de résoudre certains problèmes par des moyens techniques (surchiffrement régulier d'informations devant être protégées à long terme, horodatage et signature régulière de documents notariés...) ; cette approche est parfois indispensable mais ne peut être généralisée à cause des contraintes qu'elle impose.

Par conséquent, une analyse qui semble valable à plus de 15 ans a été développée. Les résultats présentés doivent cependant être pris avec précaution. Il suffit pour s'en convaincre de comparer l'état de l'art actuel à celui d'il y a quelques dizaines d'années.

1.4 Organisation de l'annexe

L'ensemble des règles et recommandations sont repérées selon la codification suivante : les premières lettres « Règle » ou « Recom » indiquent si l'on a affaire à une règle ou une recommandation, le domaine d'application est ensuite précisé et, finalement, un chiffre permet de distinguer les règles d'une même catégorie.

1.5 Mise à jour de l'annexe

La mise à jour de la présente annexe est réalisée par l'Agence Monégasque de Sécurité Numérique en fonction des évolutions techniques, législatives et réglementaires en matière de sécurité des systèmes d'information. Ladite mise à jour est publiée par arrêté ministériel, lequel précise les modalités de transition et date d'effet.

2 Règles et recommandations

2.1 Cryptographie symétrique

2.1.1 Taille de clé symétrique

Dans ce point sont définies les propriétés attendues de clés utilisées par des mécanismes symétriques.

Dans ce document, la taille d'une clé est le nombre de bits effectifs de cette clé, c'est-à-dire le nombre de bits réellement variables. Par exemple le DES (*Data Encryption Standard*) utilise des clés de 64 bits mais seuls 56 de ces bits peuvent être choisis aléatoirement, les 8 bits restants servant de contrôle de parité. C'est pourquoi on considère que les clés DES ont une taille de 56 bits.

Le respect des règles définies ci-dessous est une condition nécessaire qui ne peut être considérée comme suffisante. Une analyse cryptographique du mécanisme est en particulier indispensable.

RègleCléSym-1. La taille minimale des clés symétriques utilisées jusqu'en 2020 est de 100 bits.

RègleCléSym-2. La taille minimale des clés symétriques devant être utilisées au-delà de 2020 est de 128 bits.

RecomCléSym-1. La taille minimale recommandée des clés symétriques est de 128 bits.

Remarques :

- L'impact en termes de performances de l'emploi de clés d'au moins 256 bits est souvent faible, comme le montre l'exemple de l'AES.
- L'emploi de clés de 256 bits permet de s'assurer que les attaques génériques par recherche exhaustive seront inopérantes, y compris à assez long terme. Ceci ne veut bien entendu pas dire que tout mécanisme utilisant de telles clés est cryptographiquement sûr.
- L'emploi de clés de 112 bits, comme dans le cas du triple DES, ne pose pas de problème pratique de sécurité vis-à-vis d'attaques par recherche exhaustive. L'utilisation du triple DES peut cependant être déconseillée pour d'autres raisons, en particulier liées à la taille du bloc (64 bits) insuffisante pour assurer une sécurité pratique avec certains modes opératoires classiques.

2.1.2 Chiffrement symétrique

2.1.2.1 Chiffrement par bloc

Les deux caractéristiques les plus simples d'un mécanisme de chiffrement par bloc sont la taille effective de la clé ainsi que la taille des blocs traités (voir documents techniques relatifs à la cryptographie sur le site de l'AMSN : amsn.gouv.mc). Les règles et recommandations concernant la taille effective de la clé ont été présentées au point précédent.

a) Taille de bloc.

RègleBlocSym-1. La taille minimale des blocs de mécanismes de chiffrement par bloc utilisés jusqu'en 2020 est de 64 bits.

RègleBlocSym-2. Pour une utilisation au-delà de 2020, la taille minimale des blocs de mécanismes de chiffrement par bloc est de 128 bits.

RecomBlocSym-1. La taille recommandée des blocs de mécanismes de chiffrement par bloc est de 128 bits.

Remarque :

L'emploi de blocs de moins de 128 bits devrait tendre à disparaître avec l'emploi d'algorithmes modernes tels que l'AES.

b) Choix de l'algorithme.

Le choix d'un algorithme de chiffrement par bloc repose sur la prise en compte des règles et recommandations liées à la taille de la clé ainsi qu'à la taille du bloc. Au-delà de la simple considération de ces deux dimensions, il faut prendre en compte la sécurité intrinsèque apportée par le mécanisme face à des attaques plus évoluées que la simple recherche exhaustive sur la clé (cryptanalyse linéaire, différentielle...).

On considère généralement qu'une attaque est qualifiée par :

- le nombre *Nop* d'opérations de calcul nécessaires à l'attaque, une opération étant équivalente au chiffrement d'un bloc ;
- le nombre *Nbloc* de blocs à faire chiffrer ou déchiffrer afin de réaliser l'attaque ;
- la quantité *Nmem* de mémoire nécessaire, par exemple pour stocker des précalculs.

RègleAlgoBloc-1. Pour un algorithme de chiffrement ne devant pas être utilisé après 2020, aucune attaque nécessitant moins de $Nop=2^{100}$ opérations de calcul ne doit être connue.

RègleAlgoBloc-2. Pour un algorithme de chiffrement dont l'utilisation au-delà de 2020 est envisagée, aucune attaque nécessitant moins de $Nop=2^{128}$ opérations de calcul ne doit être connue.

RecomAlgoBloc-1. Il est recommandé d'employer des algorithmes de chiffrement par bloc largement éprouvés dans le milieu académique.

Remarque importante :

- Les règles ne font pas mention du nombre de blocs Nbloc à faire chiffrer ou déchiffrer afin de réaliser l'attaque, ni de la quantité de mémoire Nmem nécessaire. Ceci tient essentiellement à la volonté de ne pas trop compliquer l'énoncé de ces règles. Il conviendra, au cas par cas, de juger si l'un de ces deux paramètres est suffisamment important afin de justifier qu'un mécanisme de chiffrement par bloc est sûr, même s'il ne vérifie pas les règles RègleAlgoBloc-1 et/ou RègleAlgoBloc-2.

Mécanisme conforme au référentiel :

L'AES, tel qu'il est spécifié dans le FIPS 197, est un mécanisme de chiffrement par bloc conforme au référentiel.

Remarque :

- Le triple DES, c'est-à-dire l'utilisation du DES avec deux clés K1 et K2 en chiffrant avec K1, déchiffrant avec K2 et chiffrant de nouveau avec K1, est un algorithme de chiffrement par bloc utilisant des clés de 112 bits et des blocs de 64 bits. Le triple DES respecte donc les règles imposant des tailles de clés et de bloc minimales lorsqu'une utilisation au-delà de 2020 n'est pas envisagée. Cependant, ce mécanisme utilise une taille de bloc inférieure à la taille préconisée dans la recommandation RecomBlocSym-1. Il convient donc d'être extrêmement prudent lorsqu'on utilise ce mécanisme avec un mode opératoire de chiffrement, d'intégrité ou bien dans des protocoles de transport de clé par exemple, en particulier à cause de faible taille du bloc. De plus, le triple DES avec deux clés est vulnérable à une attaque. La complexité de cette attaque est de 280 pour 240 couples clairs-chiffrés connus et de 2100 pour 220 couples clairs-chiffrés connus. Selon le cadre d'emploi, l'utilisation du triple DES avec deux clés peut ne pas être conforme au référentiel. En particulier, le contexte d'emploi ne doit pas permettre le chiffrement avec une même clé de plus de 220 blocs de message connus d'un attaquant.

c) Mode opératoire pour le chiffrement.

Le mode opératoire pour le chiffrement permet d'assurer la confidentialité de messages de taille quelconque à partir d'une primitive de chiffrement par bloc. Un simple mécanisme de chiffrement par bloc ne permet pas d'assurer une telle fonction, en particulier à cause de sa nature fondamentalement déterministe et de la taille imposée des blocs de données traités (voir documents techniques relatifs à la cryptographie sur le site de l'AMSN : amsn.gouv.mc).

Règles et recommandations :

Le choix d'un mode opératoire de chiffrement est très dépendant de la nature des données traitées et du modèle de sécurité envisagé pour ce mécanisme. Les règles et recommandations se veulent malgré tout relativement génériques.

RègleModeChiff-1. Au sein du modèle de sécurité correspondant à l'usage du mode de chiffrement, il ne doit exister aucune attaque de complexité inférieure à $2^{n/2}$ appels de la primitive où n est la taille en bits du bloc.

RecomModeChiff-1. L'emploi d'un mode opératoire de chiffrement non déterministe est recommandé.

RecomModeChiff-2. L'utilisation d'un mode opératoire de chiffrement se fera de préférence conjointement à l'utilisation d'un mécanisme d'intégrité. Un tel mécanisme pourra être indépendant du mode de chiffrement.

RecomModeChiff-3. On utilisera de préférence des modes opératoires disposant d'une preuve de sécurité.

Mécanisme conforme au référentiel :

Le mode de chiffrement CBC utilisant une primitive de chiffrement conforme au référentiel comme l'AES et des valeurs initiales aléatoirement choisies pour chaque message et transmises en clair est un mécanisme de chiffrement symétrique conforme au référentiel (voir documents techniques relatifs à la cryptographie sur le site de l'AMSN : amsn.gouv.mc). Il est particulièrement important de garantir que les valeurs initiales sont générées dans le périmètre de sécurité du chiffrement – par exemple dans le composant sécurisé où le mode de chiffrement et la primitive sous-jacente sont implantés et non hors de ce composant – et avec un générateur d'aléa sûr. Elles ne doivent en aucun cas pouvoir être contrôlées ou prédites par un attaquant.

2.1.2.2 Chiffrement par flot

Les algorithmes de chiffrement par flot[1] constituent l'autre grande famille de mécanismes de chiffrement symétrique (voir documents techniques relatifs à la cryptographie sur le site de l'AMSN : amsn.gouv.mc).

L'algorithme dit de « one-time pad », qui se résume à une addition bit à bit du message à chiffrer avec une clé de même taille, est à part dans la classification des algorithmes de chiffrement. Il ne peut en particulier pas être considéré comme un chiffrement par flot même si ces derniers en dérivent souvent. Cet algorithme dispose d'une sécurité parfaite. Ses contraintes d'emploi sont cependant telles que son utilisation est en pratique impossible, sauf dans des cas très particuliers. Ce mécanisme nécessite en particulier d'employer une clé à usage unique aussi longue que le message à protéger, sans aucune possibilité d'une quelconque réutilisation de cette clé. En général, l'emploi du « one-time pad » repousse simplement le problème du chiffrement au niveau de la mise en accord de clé.

[1] « Stream cipher » en anglais

Choix de l'algorithme :

Avant de définir des règles liées au choix d'algorithmes de chiffrement par flot, il convient de rappeler que ces derniers ne garantissent en général aucune forme d'intégrité des messages transmis (voir remarque ci-dessus pour les modes de chiffrement par bloc). Cependant, pour un algorithme de chiffrement par flot sans rebouclage, le déchiffrement d'un message chiffré non généré par l'algorithme de chiffrement n'apporte aucune information supplémentaire susceptible de favoriser une attaque. Le modèle de sécurité généralement retenu pour l'évaluation de tels mécanismes est la connaissance par l'adversaire du flot de sortie de l'algorithme.

Il convient par ailleurs de préciser qu'il est exclusivement question des algorithmes de chiffrement par flot « dédiés », en ce sens qu'ils ont été conçus spécialement pour cet usage. Les règles ne concernent pas les autres types de générateurs pseudo-aléatoires déterministes (mode flot de chiffrement par bloc, générateurs fondés sur des problèmes difficiles, générateurs informatiquement sûrs).

RègleChiffFlot-1. Pour un algorithme de chiffrement par flot ne devant pas être utilisé après 2020, aucune attaque nécessitant moins de 2^{100} opérations de calcul ne doit être connue.

RègleChiffFlot-2. Pour un algorithme de chiffrement par flot devant être utilisé après 2020, aucune attaque nécessitant moins de 2^{128} opérations de calcul ne doit être connue.

RecomChiffFlot-1. Il est recommandé d'employer des primitives de chiffrement par bloc et non des algorithmes de chiffrement par flot dédiés. Il est ainsi possible, si les propriétés du chiffrement par flot sont requises, d'utiliser un mode opératoire par flot de chiffrement par bloc conforme au référentiel et simulant un chiffrement par flot.

RecomChiffFlot-2. En cas d'utilisation d'un algorithme de chiffrement par flot, il est recommandé d'employer des algorithmes de chiffrement par flot largement éprouvés dans le milieu académique.

Remarque importante :

Les règles ne font pas mention de la quantité de données à chiffrer ou à déchiffrer afin de réaliser l'attaque, ni de la quantité de mémoire nécessaire. Ceci tient essentiellement à la volonté de ne pas trop compliquer l'énoncé de ces règles. Il conviendra, au cas par cas, de juger si l'un de ces deux paramètres est suffisamment important afin de justifier qu'un mécanisme de chiffrement par flot est sûr même s'il ne vérifie pas les règles énoncées.

2.1.3 Authentification et intégrité de messages

Les règles sur les méthodes d'authentification et d'intégrité de messages sont très dépendantes du mécanisme choisi. Certaines règles générales peuvent cependant être émises.

RègleIntegSym-1. Les méthodes symétriques d'intégrité les plus classiques se basent sur des mécanismes de chiffrement par bloc ou de hachage. De telles primitives doivent être conformes au référentiel.

RègleIntegSym-2. Il ne doit pas exister d'attaque sur le mécanisme d'intégrité utilisant moins de $2^{n/2}$ appels à la primitive sous-jacente où n est la taille de sortie de cette primitive.

RecomIntegSym-1. On utilisera de préférence des mécanismes disposant d'une preuve de sécurité.

Remarques :

- Par confusion avec les modes opératoires de chiffrement, l'emploi de « valeurs initiales » est parfois constaté pour des mécanismes d'intégrité tels que le CBC-MAC[2] ; de graves failles de sécurité peuvent en découler.
- Il est important de prendre en considération les capacités d'un éventuel attaquant à observer des éléments d'intégrité mais également à en obtenir pour des messages de son choix, par exemple.
- De nombreux modes, tels que le CBC-MAC, ne sont sûrs que si l'on traite au plus de l'ordre de $2^{n/2}$ blocs de messages clairs, où n désigne la taille en bits du bloc. Pour un mécanisme de chiffrement utilisant des blocs de n = 64 bits, cette limite peut être rapidement atteinte.
- L'emploi de clés de taille importante ne garantit pas nécessairement une sécurité en rapport avec cette taille. La plupart des variantes du CBC-MAC construites afin d'obtenir une sécurité comparable à celle du triple DES ont ainsi été cryptanalysées au sens où leur sécurité est plus comparable à celle du DES que du triple DES (voir documents techniques relatifs à la cryptographie sur le site de l'AMSN : amsn.gouv.mc).

[2] Pour des raisons de simplification, CBC-MAC désigne le mode avec surchiffrement également connu sous le nom de CBC-MAC « retail ». (voir documents techniques relatifs à la cryptographie sur le site de l'AMSN : amsn.gouv.mc)

- Un mécanisme d'intégrité vient souvent en complément d'un mécanisme assurant la confidentialité. La composition de deux mécanismes cryptographiques n'est jamais simple et doit être réalisée avec soin. À titre d'exemple, il est possible en associant un très bon chiffrement avec un très bon algorithme d'intégrité d'obtenir un mécanisme n'assurant plus le service de confidentialité.

Mécanismes conformes au référentiel :

Le mode d'intégrité CBC-MAC « retail » utilisant l'AES comme mécanisme de chiffrement par bloc et deux clés distinctes (une pour la chaîne CBC et l'autre pour le surchiffrement dit « retail ») est conforme au référentiel (à condition, bien entendu, de ne pas utiliser de valeur initiale) (voir documents techniques relatifs à la cryptographie sur le site de l'AMSN : amsn.gouv.mc). Il est à noter que le mode CBC-MAC sans surchiffrement n'est sûr que lorsqu'il est utilisé pour des messages de taille fixe.

Le mode d'intégrité HMAC utilisant SHA-2 comme fonction de hachage est conforme au référentiel.

Mécanisme NON CONFORME au référentiel :

Le mode d'intégrité CBC-MAC « retail » recommandé ci-dessus n'est pas conforme au référentiel s'il est utilisé avec le DES comme mécanisme de chiffrement par bloc, et ce même s'il emploie deux clés distinctes. En effet, bien qu'utilisant alors 112 bits de clé, l'observation de 2^{32} MACs valides permet ensuite de retrouver ces 112 bits de clé en effectuant « seulement » de l'ordre de 2^{56} calculs de DES.

2.2 Cryptographie asymétrique

Les mécanismes de cryptographie asymétrique reposent tous sur des problèmes mathématiques difficiles, généralement issus de la théorie des nombres. L'emploi de tels types de problèmes, difficiles à résoudre pour un attaquant, est par conséquent primordial en termes de sécurité.

2.2.1 Problèmes mathématiques asymétriques

2.2.1.1 Factorisation

Le problème de la factorisation consiste à retrouver la décomposition en facteurs premiers d'un entier donné, obtenu de manière secrète par multiplication de deux nombres premiers, généralement de taille comparable. Un tel nombre composé est classiquement appelé « module ». Le problème de la factorisation est principalement utilisé par le cryptosystème RSA (*Rivest, Shamir et Adleman*). Les calculs de chiffrement et de déchiffrement RSA font intervenir deux autres données que le module, appelées « exposant public » et « exposant secret ».

RègleFact-1. La taille minimale du module est de 2048 bits, pour une utilisation ne devant pas dépasser l'année 2030.

RègleFact-2. Pour une utilisation au-delà de 2030, la taille minimale du module est de 3072 bits.

RègleFact-3. Les exposants secrets doivent être de même taille que le module.

RègleFact-4. Pour les applications de chiffrement, les exposants publics doivent être strictement supérieurs à $2^{16}= 65536$.

RecomFact-1. Il est recommandé d'employer des modules d'au moins 3072 bits, même pour une utilisation ne devant pas dépasser 2030.

RecomFact-2. Il est recommandé, pour toute application, d'employer des exposants publics strictement supérieurs à $2^{16} = 65536$.

RecomFact-3. Il est recommandé que les deux nombres premiers p et q constitutifs du module soient de même taille et choisis aléatoirement uniformément.

2.2.1.2 Le « Logarithme discret »

Le « problème dit « du logarithme discret » » est fondé sur la difficulté d'inverser l'opération d'exponentiation dans un groupe. Ce problème peut être instancié dans différentes structures et des règles et recommandations sur les choix de paramètres à utiliser pour trois d'entre elles sont :

- les corps finis GF(p) à p éléments où p est un nombre premier ;
- les groupes des points de courbes elliptiques définies sur GF(p) où p est un nombre premier ;
- les groupes des points de courbes elliptiques définies sur GF(2^n).

Bien qu'il soit possible d'instancier ce problème dans d'autres structures, nombre d'entre elles sont à proscrire : tel est notamment le cas des corps finis de petite caractéristique, et en particulier de GF(2^n). Certaines de ces autres structures ne présentent toutefois pas de faiblesses connues, mais leur sécurité doit être étudiée au cas par cas et leur emploi doit être soumis à l'avis de l'Agence Monégasque de Sécurité Numérique.

a) Logarithme discret dans GF(*p*)

Le problème du logarithme discret dans GF(*p*) est fondé sur des calculs effectués dans le corps fini à p éléments, où p est un nombre premier également appelé « module ».

RègleLogp-1. La taille minimale de modules premiers est de 2048 bits pour une utilisation ne devant pas dépasser l'année 2030.

RègleLogp-2. Pour une utilisation au-delà de 2030, la taille minimale de modules premiers est de 3072 bits.

RègleLogp-3. On emploiera des sous-groupes dont l'ordre est multiple d'un nombre premier d'au moins 200 bits.

RecomLogp-1. Il est recommandé d'employer des modules premiers d'au moins 3072 bits, même pour une utilisation ne devant pas dépasser 2030.

RecomLogp-2. Il est recommandé d'employer des sous-groupes dont l'ordre est premier (au lieu d'être multiple d'un nombre premier).

b) Le « Logarithme discret » dans les courbes elliptiques définies sur GF(*p*)

Il est également possible de définir un problème de logarithme discret dans des structures plus complexes pour lesquelles aucun algorithme plus efficace que les méthodes génériques de calcul de logarithme discret n'est connu. C'est en particulier aujourd'hui le cas des courbes elliptiques qui sont définies sur un corps de base pouvant être, en pratique, premier (GF(*p*)) ou binaire (GF(2^n)).

RègleECp-1. Pour une utilisation ne devant pas dépasser 2020, on emploiera des sous-groupes dont l'ordre est multiple d'un nombre premier d'au moins 200 bits.

RègleECp-2. Pour une utilisation au-delà de 2020, on emploiera des sous-groupes dont l'ordre est multiple d'un nombre premier d'au moins 256 bits.

RègleECp-3. En cas d'utilisation de courbes particulières faisant reposer la sécurité sur un problème mathématique plus facile que le problème générique de calcul de logarithme discret sur courbe elliptique définie sur GF(*p*), ce problème devra vérifier les règles correspondantes.

RecomECp-1. Il est recommandé d'employer des sous-groupes dont l'ordre est premier (au lieu d'être multiple d'un nombre premier).

Mécanisme conforme au référentiel :

L'emploi de la courbe FRP256v1 – définie dans le journal officiel de la République Française no 241 du 16/10/2011 et dont les paramètres peuvent librement être intégrés dans tous les produits de sécurité – est conforme au référentiel. Il en est de même de l'emploi des courbes P-256, P-384 et P-521 définies dans le FIPS 186-2 du 27/01/2000.

c) « Le Logarithme discret » dans les courbes elliptiques définies sur GF(2^n)

RègleEC2-1. Pour une utilisation ne devant pas dépasser 2020, on emploiera des sous-groupes dont l'ordre est multiple d'un nombre premier d'au moins 200 bits.

RègleEC2-2. Pour une utilisation au-delà de 2020, on emploiera des sous-groupes dont l'ordre est multiple d'un nombre premier d'au moins 256 bits.

RègleEC2-3. Le paramètre n doit être un nombre premier.

RègleEC2-4. En cas d'utilisation de courbes particulières faisant reposer la sécurité sur un problème mathématique plus facile que le problème générique de calcul de logarithme discret sur courbe elliptique définie sur GF(2^n), ce problème devra vérifier les règles correspondantes.

RecomEC2-1. Il est recommandé d'employer des sous-groupes dont l'ordre est premier (au lieu d'être multiple d'un nombre premier).

Mécanisme conforme au référentiel :

L'emploi des courbes B-283, B-409 et B-571 définies dans le FIPS 186-2 du 27/01/2000 est conforme au référentiel.

2.2.2 Chiffrement asymétrique

Toute méthode de chiffrement asymétrique s'appuie sur un problème difficile de base. Ce dernier doit donc être en accord avec le niveau de robustesse recherché. Il est de plus possible pour certains mécanismes de chiffrement de faire la preuve, éventuellement sous certaines hypothèses, que la sécurité est équivalente à celle du problème de base et pas uniquement reliée de manière heuristique. Cette approche moderne de la cryptographie permet d'atteindre un niveau d'assurance meilleur que la simple approche qui consiste à constater l'absence d'attaques connues.

RecomChiffAsym-1. Il est recommandé d'employer des mécanismes de chiffrement asymétrique disposant d'une preuve de sécurité.

Mécanisme conforme au référentiel :

Le mécanisme de chiffrement asymétrique RSAES-OAEP défini dans le document PKCS#1 v2.1 est conforme au référentiel à condition de respecter les règles RègleFact-1, RègleFact-2, RègleFact-3 et RègleFact-4.

Mécanisme NON CONFORME au référentiel :

Le mécanisme de chiffrement asymétrique RSAES, mis en œuvre selon le document PKCS#1 v1.5 n'est pas conforme au référentiel dans un contexte où il est possible d'invoquer un oracle de vérification de padding. En effet, Bleichenbacher a mis en évidence en 1998 une attaque (attaque à messages chiffrés choisis adaptative) exploitant judicieusement un tel oracle pour retrouver le message clair correspondant à un chiffré donné [Ble98].

2.2.3 Signature asymétrique

Toute méthode de signature asymétrique s'appuie sur un problème difficile qui doit être en accord avec le niveau de robustesse recherché. Il est de plus possible pour certains mécanismes de signature de faire la preuve que la sécurité est équivalente à celle du problème de base et pas uniquement reliée de manière heuristique.

Les schémas de signature utilisent en général des fonctions de hachage dont le niveau de robustesse (voir 2.3) doit bien entendu être en accord avec le niveau de robustesse souhaité pour le mécanisme de signature.

RecomSignAsym-1. Il est recommandé d'employer des mécanismes de signature asymétrique disposant d'une preuve de sécurité.

Mécanismes conformes au référentiel :

Le mécanisme de signature asymétrique RSA-SSA-PSS[3] défini dans le document PKCS#1 v2.1 est conforme au référentiel à condition de respecter les règles RègleFact-1, RègleFact-2, RègleFact-3 et RègleFact-4.

Le mécanisme de signature asymétrique ECDSA défini dans le FIPS 186-2, ainsi que le mécanisme de signature asymétrique ECKCDSA, sont conformes au référentiel lorsqu'ils utilisent la courbe FRP256v1 – définie dans le journal officiel de la République Française n° 241 du 16/10/2011 et dont les paramètres peuvent librement être intégrés dans tous les produits de sécurité – ou lorsqu'ils utilisent l'une des courbes P-256, P-384, P-521, B-283, B-409 et B-571 définies dans le FIPS 186-2.

Mécanisme NON CONFORME au référentiel :

Le mécanisme de signature asymétrique RSASSA, mis en œuvre selon le document PKCS#1 v1.5 n'est pas conforme au référentiel lorsque l'exposant public e est petit et pour un mauvais choix d'implantation des vérifications liées au padding. En effet, Bleichenbacher a mis en évidence en 2006 une attaque permettant de forger des signatures dans ce cas [Ble06].

2.2.4 Authentification d'entités et établissement de clé

Les mécanismes interactifs d'authentification d'entités et d'établissement de clé reposent en général au moins partiellement sur des mécanismes de génération d'aléa, de hachage et de chiffrement ou de signature à clé publique (voir documents techniques relatifs à la cryptographie sur le site de l'AMSN : amsn.gouv.mc). Les règles et recommandations énoncées dans les points 2.2.2, 2.2.3, 2.3 et 2.4 s'appliquent alors directement.

Ces mécanismes peuvent également faire appel à des primitives asymétriques spécifiques, telles que les schémas d'authentification « à divulgation nulle de connaissance » ou le schéma d'établissement de clé de Diffie-Hellman ; les règles et recommandations énoncées dans la section 2.2.1 s'appliquent alors aux problèmes mathématiques sur lesquels ces primitives reposent. Bien entendu, l'évaluation du niveau de robustesse global du mécanisme doit être effectuée avec soin, même si des primitives conformes au référentiel sont employées. Une attention particulière devra notamment être portée à la résistance des mécanismes d'établissement de clé aux attaques par le milieu. L'utilisation conjointe de mécanismes d'établissement de clé et d'authentification d'entité convenablement liés l'un à l'autre peut permettre de se prémunir contre des attaques de ce type. Il est également souhaitable qu'un mécanisme d'établissement de clé assure la confidentialité dans le futur ou PFS (de l'anglais perfect forward privacy) des clés symétriques temporaires (ou clés de session) qu'il permet d'établir. On peut définir informellement la propriété de PFS

[3] RSA-SSA-PSS : « RSA Signature Scheme with Appendix – Provably Secure encoding method for digital Signatures ».

comme l'impossibilité d'obtenir quelque information que ce soit sur une clé de session pour un attaquant capable d'observer et/ou perturber les échanges d'établissement de clé au cours de laquelle cette clé de session a été établie entre deux entités et d'accéder, postérieurement à la période de validité de cette clé de session, à l'ensemble des secrets d'une de ces deux entités. Une condition nécessaire pour assurer cette propriété de PFS est de recourir, pour l'établissement des clés symétriques temporaires, à un schéma reposant sur l'emploi par les deux entités de secrets éphémères effacés après utilisation. Bien entendu, les clés symétriques temporaires doivent elles-mêmes être effacées par les deux entités à l'échéance de leur période de validité (dite : fin de session).

2.3 Fonctions de hachage

Les fonctions de hachage cryptographiques doivent avoir plusieurs propriétés telles que la résistance à la recherche de « collisions » (voir documents techniques relatifs à la cryptographie sur le site de l'AMSN : amsn.gouv.mc). De telles collisions peuvent cependant toujours être trouvées au moyen d'attaques génériques fondées sur le « paradoxe des anniversaires ». Un des buts lors de la conception d'une fonction de hachage est par conséquent de faire en sorte qu'il n'existe pas de meilleure attaque. En pratique, afin de contrer les attaques fondées sur le paradoxe des anniversaires, une empreinte doit être deux fois plus longue qu'une clé symétrique pour atteindre le même niveau de robustesse.

D'autre part, les fonctions de hachage itératives sont construites autour de fonctions plus élémentaires appelées fonctions de compression. L'existence d'attaques sur ces constituants, attaques qualifiées de « partielles » ci-dessous, n'implique pas nécessairement la possibilité d'attaquer la fonction de hachage en elle-même mais trahit des défauts de conception majeurs.

RègleHash-1. Pour une utilisation ne devant pas dépasser 2020, la taille minimale des empreintes générées par une fonction de hachage est de 200 bits.

RègleHash-2. Pour une utilisation au-delà de 2020, la taille minimale des empreintes générées par une fonction de hachage est de 256 bits.

RègleHash-3. La meilleure attaque connue permettant de trouver des collisions doit nécessiter de l'ordre de $2^{h/2}$ calculs d'empreintes, où h désigne la taille en bits des empreintes.

RecomHash-1. L'emploi de fonctions de hachage pour lesquelles des « attaques partielles » sont connues est déconseillé.

Mécanisme conforme au référentiel :

Le mécanisme de hachage SHA-256 défini dans le FIPS 180-2 est conforme au référentiel.

Mécanisme NON CONFORME au référentiel :

Le mécanisme de hachage SHA-1 défini dans le FIPS 180-2 a récemment fait l'objet d'une attaque en recherche de collision. La complexité de cette attaque est estimée à 263, c'est à dire inférieure à 280. Même si cette attaque n'a pas conduit, au moment de la rédaction de ce document, au calcul d'une collision explicite, sa seule existence montre une faille sérieuse dans la sécurité de cette fonction. Le mécanisme de hachage SHA-1 n'est donc pas conforme au référentiel. Il ne respecte ni RègleHash-1, ni RègleHash-3.

2.4 Génération d'aléa cryptographique

La qualité de l'aléa (voir documents techniques relatif à la cryptographie sur le site de l'AMSN : amsn.gouv.mc) est un élément crucial pour la sécurité d'un système, que ce soit pour la génération des clés ou pour le bon fonctionnement des primitives cryptographiques.

Dans cette partie sont énoncées les règles et les recommandations applicables à un générateur d'aléa destiné à alimenter un système cryptographique de manière durable.

Un tel générateur consiste généralement en la combinaison de différentes sources d'aléa et d'une couche de retraitement algorithmique.

Plus précisément, une source d'aléa désignera un dispositif susceptible de fournir en entrée du retraitement des éléments au moins partiellement aléatoires. Une **source d'aléa** est :

- **physique** s'il s'agit d'un **générateur physique d'aléa**, c'est-à-dire d'un dispositif physique spécialement conçu pour produire des bits aléatoires en quantité (théoriquement) illimitée ;
- **systémique** si elle correspond à une accumulation d'événements partiellement imprévisibles provenant du système (par exemple le procédé d'accumulation d'aléa de /dev/random sous Linux) ;
- **importée** s'il s'agit de données secrètes parfaitement aléatoires spécialement fournies par le reste du système d'information ;

- **manuelle** s'il s'agit de données aléatoires secrètes obtenues par action intentionnelle d'un utilisateur (par exemple : frappes au clavier, mouvements de la souris...). On note que selon les cas les sources d'aléa peuvent être disponibles de manière régulière ou, au contraire, ponctuelle.

Un **retraitement algorithmique** est un mécanisme de nature cryptographique destiné à combiner différentes sources d'aléa et à garantir dans la durée la qualité de l'aléa produit.

Un **état interne** est une donnée secrète dédiée au retraitement, destinée généralement à accumuler l'entropie des sources d'aléa. Les éventuelles clés secrètes utilisées par les mécanismes cryptographiques employés par le retraitement font également partie de l'état interne.

En l'absence de source d'aléa régulière, c'est-à-dire si les sources d'aléa disponibles sont ponctuelles, on note que le retraitement s'apparente à un **générateur pseudo-aléatoire**. Il possède nécessairement un état interne et une source d'aléa ponctuelle pour l'initialiser.

Dans le cas des systèmes pouvant être mis hors tension (cas considéré par défaut dans ce qui suit), un élément de sécurité important consiste en l'utilisation d'une **mémoire non volatile**, protégée en confidentialité et en intégrité, pour stocker des données qui seront utilisées lors de l'initialisation suivante. Ces données peuvent être mises à jour par les fonctions d'initialisation et/ou d'avancement. En particulier, de telles dispositions permettent de se protéger des attaques par rejeu et de fournir suffisamment d'entropie à l'algorithme de retraitement lors de l'initialisation.

Finalement, les différents éléments constituant un générateur d'aléa cryptographique peuvent être représentés de la manière suivante (figure 1), étant entendu que tous les composants ne sont pas forcément nécessaires et que le détail des fonctionnalités représentées peut varier d'un système à un autre.

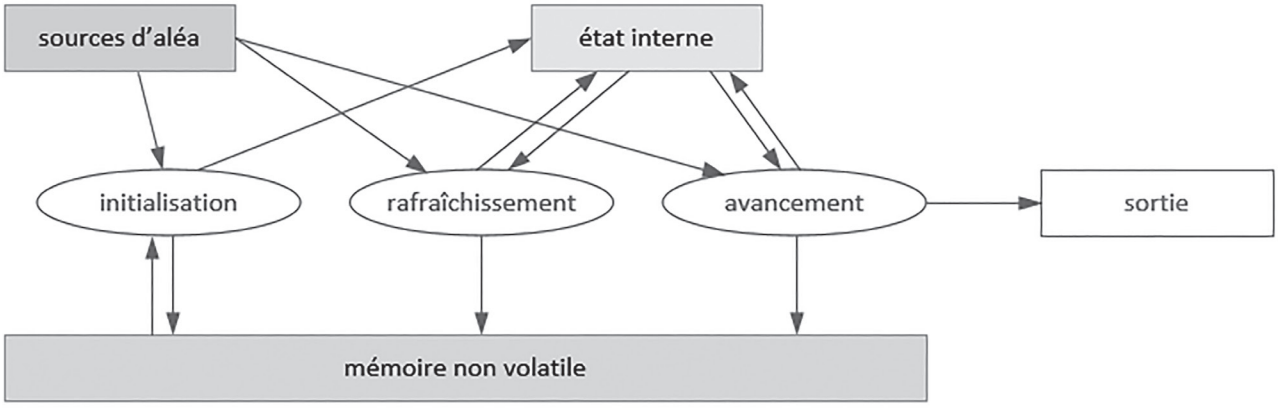

Figure 1 – Architecture générique pour la génération d'aléa cryptographique

Les règles et recommandations applicables aux générateurs d'aléa se fondent sur le constat qu'il est aujourd'hui très difficile de fournir une preuve convaincante concernant la qualité de l'aléa issu d'un générateur physique, alors qu'il est relativement aisé de se convaincre de la qualité d'un bon retraitement. Ces règles sont cependant susceptibles d'être revues si des avancées théoriques importantes sont effectuées dans le domaine des générateurs physiques d'aléa.

2.4.1 Architecture d'un générateur d'aléa

RègleArchiGDA-1. Un retraitement algorithmique disposant d'un état interne doit être employé.

RègleArchiGDA-2. En l'absence de générateur physique d'aléa, le retraitement algorithmique doit disposer d'une mémoire non volatile.

RègleArchiGDA-3. L'état interne doit être au minimum de 128 bits. En l'absence d'un rafraîchissement suffisamment fréquent par un générateur physique d'aléa, cette limite inférieure est portée à 160 bits.

RègleArchiGDA-4. La qualité des sources d'aléa ponctuelles ou régulières utilisées pour initialiser l'état interne doit être suffisante pour assurer à la valeur initiale de cet état une entropie voisine de sa longueur, ou tout au moins supérieure au seuil défini dans la règle RègleArchiGDA-3 si un raisonnement permet d'établir qu'aucune faiblesse n'en résulte.

RecomArchiGDA-1. Il est recommandé d'utiliser un retraitement avec un état interne d'au moins 256 bits, une mémoire non volatile et une source d'aléa rafraîchissant régulièrement l'état interne du générateur.

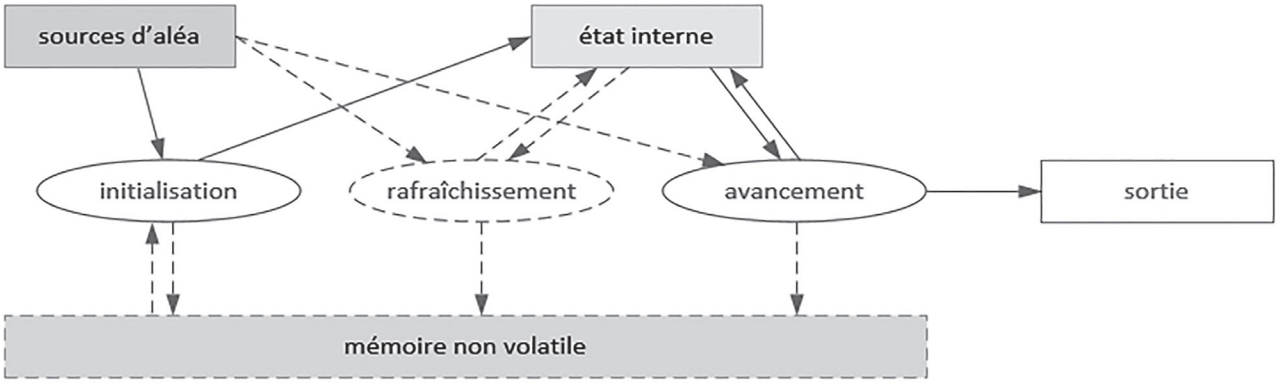

Figure 2 – Architecture minimale pour la génération d'aléa
(Les pointillés figurent les éléments recommandés.)

2.4.2 Générateur physique d'aléa

Le générateur physique d'aléa est conçu pour générer de l'aléa tout au long de la vie du système. Il importe donc de garantir autant que possible la qualité et la fiabilité de cet aléa.

RègleArchiGVA-1. Le générateur physique d'aléa doit disposer d'une description fonctionnelle. Celle-ci doit notamment indiquer les principes concourant à la génération de vrai aléa.

RègleArchiGVA-2. Des tests statistiques en sortie du générateur physique ne doivent pas faire apparaître de défauts significatifs dans l'aléa généré.

RecomArchiGVA-1. Il est souhaitable qu'un raisonnement permette de justifier la qualité de l'aléa produit par le générateur physique.

Remarque :

Les tests statistiques concernés par la règle RègleArchiGVA-2 sont des tests d'usine ou des tests ponctuels. Il ne s'agit pas d'éventuels tests de panne pouvant être réalisés en fonctionnement, au cours de l'utilisation du générateur physique.

2.4.3 Retraitement algorithmique

Les propriétés attendues du retraitement algorithmique pour la génération d'aléa sont :

RègleAlgoGDA-1. Les primitives cryptographiques employées par le retraitement algorithmique doivent être conformes au référentiel.

RègleAlgoGDA-2. Dans l'hypothèse où l'état interne est fiable, même en cas de défaillance des sources d'aléa présentes, les sorties successives du retraitement doivent être parfaitement aléatoires du point de vue de l'attaquant. De plus, la connaissance de ces sorties ne doit pas mettre en danger la confidentialité des états internes ni des sources d'aléa (fiables).

RègleAlgoGDA-3. En cas de compromission « simple » affectant ou bien l'état interne ou bien les sources d'aléa éventuellement présentes mais n'affectant pas simultanément ces deux types d'éléments, la sortie courante ne doit donner à l'attaquant aucune information exploitable sur les sorties passées.

2.5 Gestion de clés

La gestion des clés est traitée dans le document « Gestion des clés cryptographiques – Règles et recommandations concernant la gestion des clés utilisées dans des mécanismes cryptographiques ».

imprimé sur papier PEFC
IMPRIMERIE GRAPHIC SERVICE
GS COMMUNICATION S.A.M. MONACO