Groupe de travail Réseau
Request for Comments : 4772
Catégorie : Information
Traduction Claude Brière de L'Isle

S. Kelly
Aruba Networks

décembre 2006
octobre 2007

Implications pour la sécurité de l'utilisation de la norme de chiffrement des données (DES)

Statut du présent mémoire
Le présent mémo donne des informations pour la communauté de l'Internet. Il ne spécifie aucune sorte de norme Internet. Sa distribution n'est soumise à aucune restriction.

Résumé
La norme de chiffrement des données (DES, *Data Encryption Standard*) est susceptible de se prêter à des attaques en force, qui sont à la portée d'un adversaire ayant des moyens financiers modestes. Il en résulte que DES a été déconseillé, et remplacé par la norme de chiffrement évolué (AES, *Advanced Encryption Standard*). Néanmoins, de nombreuses applications continuent de s'appuyer sur DES pour la sécurité, et des concepteurs et des développeurs continuent de la prendre en charge dans de nouvelles applications. Bien que ce ne soit pas toujours inapproprié, ce l'est fréquemment. La présente note discute en détail des implications de DES sur la sécurité, de sorte que les concepteurs et les développeurs aient toutes les informations nécessaires pour prendre des décisions judicieuses concernant son utilisation.

1 Introduction

La norme de chiffrement des données [DES, *Data Encryption Standard*] est le premier algorithme de chiffrement approuvé par le gouvernement américain pour la mise à disposition du public. Les attaques en force sont immédiatement devenues un sujet de spéculation après la mise à disposition du public, et un certain nombre de chercheurs ont publié des discussions sur la faisabilité d'une attaque et les méthodologies d'une attaque en force explicite, dont la première est [DH77].

Au début du milieu des années 1990, de nombreux autres articles sont apparus, parmi lesquels celui de Wiener "Recherche efficace de clés DES" [WIEN94], et "Longueurs minimales de clé de chiffrement symétrique pour fournir une sécurité commerciale adéquate" [BLAZ96]. Alors que ces articles et divers autres discutaient les aspects théoriques de la machinerie de viol du DES, aucun ne décrivait la mise en œuvre spécifique d'une telle machine. En 1998, la Fondation de la frontière électronique (EFF, *Electronic Frontier Foundation*) est allée plus loin, en construisant réellement un appareil et en en publiant librement les détails de mise en œuvre [EFF98].

En dépit du fait que l'EFF démontrait clairement que DES pouvait être forcé dans une moyenne d'environ 4,5 jours avec un investissement de moins de 250 000 $ en 1998, beaucoup ont encore alors continué de s'appuyer sur cet algorithme, plus de 8 ans après. Aujourd'hui, le paysage est significativement différent : DES peut être cassé par une large gamme d'attaquants en utilisant des technologies qui n'étaient pas disponibles en 1998, y compris des circuits intégrés prédiffusés programmables (FPGA, *Field Programmable Gate Array*) et des réseaux zombies [BOT05] bon marché. Elles sont, avec d'autres méthodologies d'attaque, décrites en détail ci-après.

Étant donné que la norme de chiffrement évolué [AES, *Advanced Encryption Standard*] a été approuvée par le gouvernement américain (selon certains scénarios d'utilisation) pour des applications très secrètes [AES-NSA], et que le triple DES (3DES) n'est pas susceptible de ces mêmes attaques, on est fondé à se demander s'il faut encore s'occuper de DES. Dans des circonstances idéales, on pourrait simplement s'en passer, mais malheureusement, ce n'est pas aussi simple que cela aujourd'hui. DES a connu un grand développement depuis sa publication dans les années 1970, et de nombreux systèmes s'appuient aujourd'hui sur elle. Le remplacement en bloc de tels systèmes serait très coûteux. Une approche plus réaliste est celle d'un remplacement graduel de ces systèmes, et cela implique une durée indéfinie de prise en charge de la rétrocompatibilité.

En plus de la rétrocompatibilité, il peut y avoir dans des instances isolées des arguments valides pour la continuation de la prise en charge de DES. Toujours est-il que s'appuyer sur cet algorithme déconseillé est une sérieuse erreur du point de vu de la conception de la sécurité dans de nombreux cas. La présente note vise à clarifier les implications de ce choix sur la sécurité étant donné l'état de la technologie d'aujourd'hui, de sorte que les développeurs puissent prendre des décisions en connaissance de cause sur la mise en œuvre ou non de cet algorithme.

1.1 Résumé des conclusions et recommandations

Depuis maintenant de nombreuses années, l'utilisation de DES a été activement déconseillée par la communauté de la sécurité de l'IETF, mais nous continuons néanmoins à la voir utiliser. Étant donné le grand nombre d'attaques réelles parues dans la presse et les conseils diffusés pour en décourager l'usage, une question se pose : pourquoi les gens n'écoutent-ils pas ? On ne peut que faire des spéculations, mais une possibilité serait qu'ils ne comprennent tout simplement pas à quel point DES a été marginalisée par l'avancement des sciences et techniques cryptographiques. Une autre possibilité est que nous n'ayons pas été suffisamment explicites et agressifs à son sujet. Gardant ces deux possibilités à l'esprit, la présente note va essayer de dissiper les dernières illusions restantes.

La profondeur des connaissances de base nécessaire pour vraiment comprendre et apprécier pleinement les risques que présente aujourd'hui l'utilisation de DES est quelque peu décourageante, et un large survol de la littérature suggère qu'il y a très peu de documents publiés qui mettent en application plus qu'une fraction de tous les éléments à considérer en une seule fois, à l'exception notable de [CURT05]. Cependant, même ce travail ne rassemble pas tous les morceaux du puzzle d'une façon qui informe vraiment un développeur des risques réels encourus, aussi allons nous essayer de boucher tous les trous restants.

Par commodité, le paragraphe suivant contient un bref résumé des recommandations. Si vous ne connaissez pas la position actuelle de l'IETF sur DES, et que tout ce que vous vouliez est un résumé, vous pouvez vous contenter de lire simplement le paragraphe de résumé des recommandations, et sauter le reste du document. Si vous voulez une vision plus détaillée de l'histoire et de l'état de l'art actuel par rapport aux attaques contre DES, vous trouverez cela dans les sections suivantes.

1.1.1 Résumé des recommandations

Il y a plusieurs façons d'attaquer un algorithme cryptographique, de la simple attaque en force (en essayant chaque clé jusqu'à trouver la bonne) à des approche plus subtiles d'analyse cryptographique, qui tiennent compte de la structure interne du chiffre. Comme noté dans l'introduction, un système dédié capable de casser en force des clés DES en moins de 5 jours a été créé en 1998. Les estimations courantes de la "Loi de Moore" suggèrent qu'une machine similaire pourrait aujourd'hui être construite pour environ 15 000 $ ou moins, et pour le coût du système original (~ 250 000 $) on pourrait probablement construire une machine capable de casser les clé DES en quelques heures.

De plus, il y a eu un certain nombre d'attaques réparties réussies contre DES [CURT05], et avec l'arrivée récente de réseaux zombies *(botnet)* [BOT05], ces résultats sont d'autant moins onéreux. De plus, il y a un certain nombre d'attaques cryptanalytiques contre DES, et si une bonne part d'entre elles restent jusqu'à présent purement théoriques, au moins une a récemment été mise en œuvre en utilisant un FPGA qui peut déduire une clé DES en 12 à 15 heures [FPL02]. En clair, DES ne peut pas être considéré comme un algorithme cryptographique "fort" selon les normes actuelles.

Pour résumer les recommandations actuelles sur l'utilisation de DES, la simple réponse est "ne l'utilisez pas – il n'est pas sûr." Bien qu'il y ait des cas d'utilisation pour lesquels la sécurité de DES serait suffisante, ils exigent normalement un expert en sécurité pour déterminer quand c'est vrai. Et aussi, il y a beaucoup plus d'algorithmes de sécurité disponibles aujourd'hui (par exemple, 3DES, AES) qui sont des choix beaucoup plus sûrs. Le seul cas général dans lequel DES pourrait toujours être accepté est quand il est absolument exigé pour la rétrocompatibilité, et quand le coût de mise à niveau outrepasse le risque exposé. Cependant, même dans ces cas, la recommandation devrait probablement être d'éliminer de tels systèmes.

Si vous êtes simplement intéressés par les recommandations actuelles, vous l'avez : n'utilisez pas DES. Si vous vous intéressez à la façon dont on arrive à cette conclusion, lisez ce qui suit !

2 Pourquoi le chiffrement ?

Pour évaluer les implications de l'utilisation de DES sur la sécurité, il est utile et instructif de réviser les raisons de base de l'utilisation du chiffrement. En général, on chiffre des informations dont on désire assurer la confidentialité. C'est à dire qu'on veut limiter l'accès à des informations pour garder quelque chose privé ou secret. Dans certains cas, on veut partager les informations au sein d'un groupe limité, et dans d'autres cas, on peut vouloir être le seul détenteur des informations en question.

Parfois, les informations qu'on veut protéger n'ont de valeur que pour l'individu (par exemple, un agenda), et la perte de confidentialité, bien que potentiellement dommageable de façon limitée, ne serait normalement pas catastrophique. Dans d'autres cas, les informations peuvent avoir des implications financières significatives (par exemple, le plan de marketing stratégique d'une entreprise). Et dans d'autres, des vies peuvent être en jeu.

Pour évaluer nos exigences de confidentialité en termes de force du chiffrement, nous devons évaluer les informations que nous essayons de protéger, à la fois pour nous et pour un attaquant potentiel. Diverses métriques peuvent être utilisées pour ce faire :

- Coût de la perte de confidentialité : Que peut on perdre si un adversaire devait découvrir notre secret ? Cela donne une certaine mesure des efforts que l'on accepterait de faire pour protéger le secret.
- Valeur pour l'adversaire : Que l'attaquant peut-il gagner à découvrir notre secret ? Cela donne une mesure de ce qu'un adversaire sera raisonnablement prêt à dépenser pour découvrir le secret.
- Fenêtre d'opportunité : Pendant combien de temps les informations ont-elles de la valeur pour un adversaire ? Cela donne une mesure dans laquelle une faiblesse peut être acceptable. Par exemple, si les informations sont précieuses pendant des mois pour un attaquant et que cela ne prend que quelques jours pour casser le codage, on a probablement besoin d'un codage plus fort. D'un autre côté, si la fenêtre d'opportunité se mesure en secondes, un algorithme de chiffrement qu'il faut des jours pour casser peut être acceptable.

Il y a certainement d'autres facteurs à considérer quand on fait une analyse de sécurité complète, mais ceci suffit à donner un sens général aux importantes questions auxquelles il faut répondre en évaluant la candidature de DES comme algorithme de chiffrement.

3 Réalité des applications et des menaces

De nombreuses applications couramment utilisées s'appuient sur le chiffrement pour assurer la confidentialité sur l'Internet d'aujourd'hui. Pour évaluer si un algorithme cryptographique donné est suffisant dans ce contexte, nous devons commencer par nous poser quelques questions de base : quels sont les risques réels pour ces applications, c'est-à-dire, quelle est la probabilité qu'une application puisse réellement être attaquée, par qui, et pour quelles raisons ?

Alors qu'il est difficile d'arriver à une réponse passe partout fondée sur des descriptions d'application générales, on peut facilement avoir une opinion sur la réalité des menaces sur beaucoup de ces applications. Il est important de noter que ce qui suit n'est pas une récapitulation exhaustive de toutes les menaces et attaques vraisemblables, mais plutôt, un échantillon qui illustre que les menaces réelles sont plus présentes que ce que l'intuition nous suggère.

Voici quelques exemples d'applications courantes et des menaces qui s'y rapportent :

- Réseau virtuel de site à site : Souvent, ils sont utilisés pour connecter les bureaux séparés géographiquement d'une entreprise. Les données qui traversent de telles liaisons sont souvent critiques pour les affaires et parfois hautement confidentielles. Le FBI estime que chaque année, des millions de dollars U.S. sont perdus au profit de concurrents étrangers qui espionnent délibérément l'industrie et les technologies des USA [FBI06]. Une recherche dans Google sur 'espionnage industriel' donne de nombreux liens intéressants, dont certains indiquent que les concurrents étrangers ne sont pas la seule menace pour l'économie américaine. Visiblement, cette menace peut être généralisée pour y inclure les économies de tous les pays.
- Accès réseau distant pour les affaires : Voir le point précédent.
- Codage de messagerie électronique : Voir Réseau de paquets virtuels de site à site.
- Banque en ligne : Actuellement, la menace la plus commune sur la banque en ligne est dans la forme du "hameçonnage", qui ne consiste pas à casser le chiffrement d'une session, mais vise à tromper les usagers en leur fournissant des informations sur leur compte. En général, les attaques directes sur le chiffrement de session ne vont pas très loin pour cette application. Cependant, si une banque particulière était connue pour utiliser un algorithme de chiffrement faible pour la sécurité des sessions, il pourrait devenir rentable de développer une plus grande attaque contre cette banque. Étant donné que des membres du crime organisé ont été découverts derrière de nombreuses attaques d'hameçonnage, il n'est pas difficile d'imaginer de tels scénarios.
- Transferts électroniques de fonds (EFT) : La capacité à répéter ou à modifier des EFT légitimes a des incitations financières (et des implications) évidentes. Et aussi, un espion industriel peut trouver un grand nombre de renseignements de valeur dans les transactions financières de l'entreprise cible.
- Les achats en ligne (commerce électronique) : Le FBI a enquêté sur un grand nombre d'attaques organisées sur des applications de commerce électronique [FBI01]. Si un attaquant a la capacité de surveiller le trafic de commerce électronique dirigé sur un gros négociant qui s'appuie sur un chiffrement faible, l'attaquant pourrait engranger une grande quantité d'informations sur le crédit des consommateurs. C'est le genre de données que les "hameçonneurs" récoltent habituellement sur une beaucoup plus petite échelle, aussi peut on facilement imaginer la valeur d'une telle cible.
- Applications de voix sur IP fondées sur l'Internet (par exemple, Skype) : Alors que de nombreuses utilisations de cette technologie sont inoffensives (par exemple, les appels longue distance aux membres de la famille), la technologie VoIP est aussi utilisée pour des besoins professionnels (voir la discussion des estimations du FBI concernant l'espionnage économique ci-dessus).
- Téléphonie cellulaire : Les téléphones cellulaires sont très courants, et sont fréquemment utilisés pour des conversations confidentielles, dans les affaires, en médecine, en matière de police , et pour d'autres applications.
- LAN sans fil : Les technologies sans fil sont utilisées dans de nombreux secteurs, y compris la Bourse de New York [NYSE1]. Les incitations financières sont significatives dans certains cas pour un attaquant.
- Communications personnelles (par exemple, messagerie instantanée sécurisée) : De telles communications peuvent être utilisées pour des communications d'affaires (voir l'exposé sur l'espionnage industriel ci-dessus), et peuvent aussi être utilisées pour des applications financières telles que des opérations de bourse ou d'obligations. Ceci a à la fois des

implications d'espionnage économique/industriel et financières.
- o Chiffrement de disque dur d'ordinateur portable : Voir la discussion ci-dessus sur l'espionnage économique/ industriel. Considérer aussi que les ordinateurs portables volés et perdus ont été cités comme une des pertes de contrôle les plus significatives sur des informations personnelles sensibles de ces dernières années, en particulier l'affaire de la perte des données sur les anciens combattants [VA1].

Il y a des menaces réelles sur des applications de chiffrement de tous les jours, dont certaines pourraient être très lucratives pour un attaquant (et par extension, très coûteuses pour la victime). Il est important de noter que si certaines de ces attaques ne sont pas fréquentes aujourd'hui, c'est précisément parce que la menace est connue, et que des algorithmes cryptographiques d'une force appropriée sont utilisés. Si des algorithmes cryptographiques "faibles" étaient utilisés à la place, leurs implications donneraient sans doute à penser.

En restant dans les objectifs du présent document, il est important de noter que le gouvernement U.S. n'a jamais approuvé l'utilisation de DES pour autre chose que des applications non confidentielles. Alors que DES est toujours approuvé pour des utilisations non confidentielles jusqu'au 19 mai 2007, le gouvernement U.S. a vu clairement le besoin de passer à un niveau supérieur. Pour des détails sur le plan de transition du DES de l'Institut National des normes et technologies (NIST), voir [NIST-TP]. En dépit de ce fait, DES est encore parfois choisi pour protéger certaines des applications décrites ci-dessus. Nous allons ci-après exposer pourquoi il faut, dans de nombreux cas, y remédier.

4 Attaquer DES

DES est un bloc de chiffrement de 64 bits qui a une taille de clé de 56 bits. La clé a en fait 64 bits, ce qui correspond à la taille du bloc), mais un bit dans chaque octet est utilisé comme bit de 'parité', et ne sert pas au chiffrement. Un exposé complet de l'histoire de la norme DES avec une description accessible de l'algorithme figure dans [SCHN96].

Une description détaillée des divers types d'attaques contre les algorithmes cryptographiques sort du domaine d'application du présent document, mais pour être précis, nous donnerons les brèves descriptions suivantes. Il y a deux aspects généraux de l'attaque qui doivent être considérés : la forme de l'entrée/sortie avec la façon dont nous pouvons l'influencer, et la fonction interne des opérations de chiffrement elles-mêmes.

En termes de formes des entrées/sorties, certaines des caractéristiques d'attaque les plus couramment exposées sont les suivantes :
- o texte clair connu – l'attaquant connaît une partie du texte en clair qui correspond à une partie d texte chiffré,
- o seulement du texte chiffré – seul le texte chiffré est disponible à l'attaquant, qui a peu ou pas d'informations sur le texte en clair,
- o texte en clair choisi – l'attaquant peut choisir le texte clair qui est chiffré, et obtenir le texte chiffré correspondant,
- o attaques de jour anniversaire – s'appuient sur le fait que pour N éléments, des collisions sont prévisibles dans ~sqrt(N) échantillons aléatoires ; pour les systèmes utilisant le mode CBC avec des vecteur d'initialisation (IV) aléatoires, les collisions de texte chiffré sont prévisibles dans environ 2^28 échantillons. De telles collisions laissent échapper des informations sur les textes en clair correspondants : si la même clé cryptographique est utilisée, le OUx des IV est égal au OUx des textes en clair,
- o attaques par rencontre en chemin – moyenner les caractéristiques de jour anniversaire pour pré-calculer les valeurs potentielles de collision de clé.

Du fait de la portée limitée du présent document, ce sont de très brèves descriptions de sujets très complexes. Pour un exposé détaillé sur ces questions et ce qui s'y rapporte, voir [SCHN96], [HAC], ou [FERG03].

Comme pour les caractéristiques d'attaques se rapportant aux aspects opérationnels des algorithmes de chiffrement, il y a essentiellement deux grandes classes à considérer : les attaques par analyse du chiffrement, qui exploitent la structure ou fonction interne de l'algorithme de chiffrement, et les attaques en force, dans lesquelles l'attaquant essaye systématiquement les clés jusqu'à trouver la bonne. On peut aussi les désigner autrement sous les noms respectifs d'attaque de boîte blanche et d'attaque de boîte noire. Nous allons exposer ceci plus en détail ci-dessous.

4.1 Attaques en force

En général, une attaque en force consiste en l'essai de chaque clé possible jusqu'à trouver la clé correcte. Dans le pire cas, ceci exigera 2^n étapes pour une taille de clé de n bits, et en moyenne, cela exigera 2^n-1 étapes. Pour DES, cela implique 2^56 opérations de chiffrement dans le pire des cas, et 2^55 opérations de chiffrement en moyenne, si nous supposons qu'il existe des raccourcis. Comme elle se présente, la propriété de complémentation de DES fait une attaque qui donne une réduction d'un facteur 2 pour une attaque de texte en clair choisi, et donc cette attaque exige une moyenne de 2^54 opérations de chiffrement.

Ci-dessus nous nous référons à 2^n 'étapes' ; noter que ce qu'englobe une 'étape' dépend dans une certaine mesure du premier aspect de l'attaque décrit ci-dessus, c'est-à-dire, quelle influence et quelle connaissance nous avons des formes d'entrée/sortie. Rappelez vous que dans le pire cas, nous effectuerons 72 057 594 037 927 936 – plus de 72 quadrillions –

de ces 'étapes'. Dans le cas le plus difficile, nous n'avons que le texte chiffré, et aucune connaissance de l'entrée, et c'est très important.

Si l'entrée est effectivement aléatoire, on ne peut pas dire, simplement en regardant un bloc déchiffré, si on a réussi ou non. Il ait possible qu'on doive avoir recours à d'autres calculs potentiellement coûteux pour le déterminer. Alors que l'effet de tout calcul supplémentaire sera linéaire par rapport à toutes les clés, la répétition d'une grande quantité de calculs supplémentaires jusqu'à 72 quadrillions de fois pourrait avoir un impact significatif sur le coût d'une attaque en force contre l'algorithme. Par exemple, si cela prend une microseconde supplémentaire par calcul, cela va ajouter presque 101 jours à notre pire cas de temps de recherche, en supposant une recherche de clé en série.

D'un autre côté, si on peut contrôler l'entrée de la fonction de chiffrement (texte clair connu), on sait précisément ce qu'on attend de la fonction de déchiffrement, aussi on sait tout de suite détecter qu'on a trouvé la clé. Autrement, même si on ne connaît pas l'entrée exacte, si on sait quelque chose sur elle (par exemple, que c'est de l'ASCII), avec une quantité limitée de calculs supplémentaires, on peut inférer qu'on a très vraisemblablement trouvé une clé. Visiblement, savoir laquelle de ces conditions est vérifiés peut influencer significativement le temps d'attaque.

4.1.1 Attaques parallèles et distribuées

Étant donné qu'une attaque en force implique d'essayer systématiquement les clés jusqu'à trouver la bonne, elle est un bon candidat évident pour la parallélisation. Si on a N processeurs, on peut en gros trouver la clé N fois plus vite que si on a qu'un seul processeur. Cela exigé une sorte d'entité de contrôle centralisée qui distribue le travail et surveille le processus de recherche, mais est d'une mise en œuvre présentant peu de difficultés.

Il y a au moins deux approches de la parallélisation d'une attaque en force sur un bloc chiffré : la première est de construire un matériel à grande vitesse spécialisé qui puisse rapidement passer les clés en revue tout en effectuant les opérations cryptographiques et de comparaison, puis dupliquer ensuite de nombreuses fois le matériel, tout en fournissant le contrôle centralisé. La seconde implique d'utiliser de nombreuses copies d'un matériel tout venant (par exemple, un micro ordinateur), et de distribuer la charge entre eux tout en les plaçant sous le contrôle d'un ou plusieurs systèmes centraux. Ces approches sont toutes deux exposées plus en détail aux sections 5 et 6.

4.2 Attaques cryptanalytiques

Les attaques en force sont appelées ainsi parce qu'elles n'exigent pas beaucoup d'intelligence dans le processus d'attaque – elle essayent simplement une clé après l'autre, avec peu ou pas d'élagage intelligent de l'espace de clés. D'un autre côté, les attaques cryptanalytiques s'appuient sur l'application d'une certaine intelligence à l'avance, et ce faisant, fournissent une réduction significative de l'espace de recherche.

Bien qu'une discussion en profondeur des techniques cryptanalytiques et des attaques qui en résultent soit bien en dehors du domaine d'application du présent document, il est important de dire un mot de cette question afin d'établir les bases de la suite de l'exposé. Il est aussi important de noter que, en général, l'analyse cryptographique peut s'appliquer à tout algorithme cryptographique avec des niveaux de succès variés. Cependant, nous nous confinerons ici à exposer les résultats spécifiques de DES.

Voici un très bref résumé des attaques cryptanalytiques actuellement connues contre DES :

- Cryptanalyse différentielle – Exposée pour la première fois par Biham et Shamir, cette technique (dit très simplement) analyse comment les différences dans le texte en clair correspondent aux différences dans le texte chiffré. Pour les détails, voir [BIH93].
- Cryptanalyse linéaire – Décrite pour la première fois par Matsui, cette technique utilise des approximations linéaires pour décrire les fonctions internes de DES. Pour les détails, voir [MAT93].
- Attaque par interpolation – Cette technique représente les S-boxes de DES par des fonctions algébriques, puis estime les coefficients des fonctions. Pour plus d'informations, voir [JAK97].
- Attaque de collision de clés - Cette technique exploite le paradoxe de l'anniversaire pour produire des collisions de clés [BIH96].
- Analyse différentielle des fautes - Cette attaque exploite les caractéristiques électriques de l'appareil de chiffrement, en induisant sélectivement des fautes et en comparant le résultat avec des sorties non influencées. Pour des précisions, voir [BIH96-2].

Actuellement, les attaques cryptanalytiques les plus connues du public contre DES sont l'analyse cryptographique linéaire et différentielle. Ces attaques ne sont en général pas considérées comme praticables, car elles exigent respectivement 2^43 et 2^47 de paires de texte en clair/ texte chiffré connues. Pour avoir une idée de ce que cela signifie en pratique, considérons ce qui suit :

- Pour la cryptanalyse linéaire (la plus efficace des deux attaques), l'attaquant doit pré calculer et mémoriser 2^43 textes chiffrés ; cela exige 8 796 093 022 208 (presque 9 trillions) d'opérations de chiffrement.
- Chaque bloc de texte chiffré est de 8 octets, ainsi la mémoire totale nécessaire est de 70 368 744 177 664 octets, ou

environ 70 369 gigaoctets de mémoire. Si les blocs de texte en clair ne peuvent pas être déduits automatiquement, ils doivent eux aussi être mémorisés, ce qui double potentiellement les besoins de mémoire.

- Les 2^43 blocs de texte en clair connus doivent d'une certaine façon être injectés dans l'appareil attaqué, et cet appareil ne doit pas changer la clé de chiffrement pendant ce temps.

En clair, cette attaque pose des problèmes pratiques. Cependant, il est frappant de voir à quel point 70 000 giga octets de mémoire est bien moins impressionnant que ce que cela paraissait en 1993, quand Matsui a proposé cette attaque. Aujourd'hui, on peut avoir 400 GO de disque dur pour environ 0,5 $/giga octet. Si on a seulement besoin de mémoriser le texte chiffré connu, cela monte à ~176 disques durs à un coût de moins de 25 000 $. C'est probablement praticable avec la technologie d'aujourd'hui pour un adversaire ayant des ressources financières significatives, alors que c'était difficile à imaginer en 1993. Ceci dit, il reste de nombreux autres problèmes pratiques.

4.3 Considérations pratiques

Ci-dessus, nous décrivons plusieurs types d'attaques contre DES, dont certaines d'entre elles sont plus praticables que d'autres, mais il est très important de reconnaître que la force brute représente le plus mauvais cas, et les attaques d'analyse cryptographiques ne peuvent que s'améliorer sur ce point. Si une attaque en force contre une application DES donnée est réellement faisable, se soucier de la praticabilité des autres modes d'attaque théoriques est une simple distraction. La ligne rouge est celle-ci : si DES peut être cassé en force à un coût que l'attaquant peut s'offrir aujourd'hui, ce coût va immanquablement baisser avec l'avancée de la technologie.

5 Le casseur de DES d'EFF

Sur la question de savoir si DES est susceptible d'attaques en force d'un point de vue pratique, la réponse est résolument et sans équivoque "oui". En 1998, la fondation pour la frontière électronique a financé la construction d'un "casseur de DES", et a ensuite publié "Casser DES" [EFF98]. Pour un coût inférieur à 250 000 $, ce système peut trouver une clé DES de 56 bits en environ neuf jours dans le plus mauvais cas, et en 4,5 jours en moyenne.

Citation tirée de [EFF98],
"La conception du casseur de DES d'EFF est d'un concept simple. Elle consiste en un ordinateur personnel ordinaire connecté à un grand dispositif de processeurs du commerce. Le logiciel du micro ordinateur donne pour instruction aux processeurs du commerce de commencer à chercher, et interagit avec l'usager. Le processeur tourne sans autre aide de la part du logiciel jusqu'à ce qu'il trouve une clé potentiellement intéressante, ou ait besoin de directives pour chercher dans une nouvelle partie de l'espace de clés. Le logiciel interroge périodiquement les processeurs pour collecter toutes les clés potentiellement intéressantes qu'ils ont pu trouver.

Le travail du matériel n'est pas de trouver la réponse, mais plutôt d'éliminer la plupart des réponses qui sont incorrectes. Le logiciel est alors assez rapide pour rechercher les clés restantes potentiellement correctes, puis passe au crible les fausses réponses positives pour dégager la réponse réelle. La force de la machine est qu'elle duplique des milliers de fois un circuit de recherche simple mais utile, permettant au logiciel de trouver les réponses en cherchant seulement sur une minuscule fraction de l'espace de clés.

Tant qu'il y a un petit bout de logiciel pour coordonner les efforts, le problème de la recherche d'une clé DES est 'hautement parallélisable'. Cela signifie que le problème peut utilement être résolu par de nombreuses machines travaillant en parallèle, simultanément. Par exemple, un seul processeur casseur de DES pourrait trouver une clé en cherchant pendant de nombreuses années. Un millier de processeur casseurs de DES peuvent résoudre le même problème dans le millième du temps. Un million de processeurs casseurs de DES pourraient théoriquement résoudre le même problème dans environ le millionième du temps, bien que la redondance du démarrage de chaque processeur deviendrait alors visible dans le temps requis. La machine réelle construite contient 1536 processeurs."

Ce projet démontre clairement qu'un système praticable d'attaque en force contre DES est tout à fait à la portée de beaucoup plus que ce qu'on supposait précédemment. Pratiquement tous les gouvernements du monde peuvent facilement produire une telle machine, et en fait, beaucoup d'hommes d'affaires le pourraient. Et ceci était en 1998 ; les avancées technologiques effectuées depuis ont largement réduit le coût d'un tel appareil. Ceci est exposé ci-après.

6 Autres projets de casse de DES

Au milieu des années 1990, beaucoup s'intéressaient à savoir si on pouvait en pratique ou non casser DES. RSA sa financé une série de défis DES sur une période de trois ans commençant en janvier 1997. Ces défis ont été créés afin d'aider à souligner que les limitations de la force cryptographique imposées par le gouvernement américain sur les politiques d'exportation étaient bien trop modestes pour satisfaire aux exigences de sécurité de la plupart des utilisateurs.

Le premier défi DES fut résolu par le groupe DESCHALL, conduit par Rocke Verser, Matt Curtin, et Justin Dolske

[CURT05][RSA1]. Ils ont créé un groupe faiblement lié conduit par des volontaires et appuyé par des universités et des entreprises du monde entier qui ont fait don de leurs cycles de CPU non utilisés au groupe. Ils ont trouvé la clé en 90 jours.

Le second défi DES fut annoncé le 19 décembre 1997 [RSA2][CURT05], et le 26 février 1998, RSA annonçait un gagnant. Cette fois, le défi avait été relevé par un groupe appelé distributed.net travaillant avec l'EFF, en un total de 39 jours [RSA3] [CURT05]. Ce groupe coordonnait 22 000 participants et plus de 50 000 CPU.

Le troisième défi DES fut annoncé le 22 décembre 1998 [RSA4][CURT05], et le 19 janvier 1999, RSA annonçait le gagnant. Cette fois encore, le défi fut relevé par distributed.net travaillant avec l'EFF, en un total de 22 heures [RSA5]. C'était une amélioration saisissante par rapport au second défi, et devrait donner une bonne idée de ce qui nous attend avec DES.

7 Construire aujourd'hui un casseur de DES

Nous avons vu ce qui a été fait à la fin des années 1990 – qu'en est-il aujourd'hui ? Un survol de la littérature peut nous conduire à conclure que ce sujet n'intéresse plus les cryptographes. Et donc, il ne nous reste qu'à déduire les possibilités fondées sur les technologies disponibles actuellement. Une façon d'avoir une approximation est d'appliquer une variation de la "Loi de Moore" : Supposons que le coût d'un appareil comparable à celui construit par l'EFF soit en gros divisé par deux tous les N mois. Si nous prenons N=18, alors pour un appareil coûtant 250 000 $ à la fin de 1998, cela donnerait une prédiction de la courbe suivante :

- mi 2000 : 125 000 $
- début 2002 : 62 500 $
- mi 2003 : 31 250 $
- début 2006 : 15 625 $

Il est important de noter que strictement parlant, la "Loi de Moore" est plus une approximation informelle qu'une loi, bien qu'il est été prouvé qu'elle est d'une étrange précision sur les dernières 40 années environ. On peut aussi ne pas être d'accord avec l'utilisation de l'intervalle de 18 mois, et préférer à la place une période plus prudente de 24 mois. Aussi, ces chiffres devraient être pris avec le proverbial grain de sel. Toujours est-il qu'il est important de reconnaître que ceci n'est pas le coût pour casser une clé, mais celui de l'entrée dans le métier de casseur de clé. Offrir des services de cassage de clé et tenir la machine relativement occupée diminuerait considérablement le coût jusqu'à quelques centaines de dollars par unité ou moins.

Étant donné que de tels calculs tiennent en gros pour d'autres technologies de calcul sur le même intervalle de temps, l'estimation ci-dessus ne semble pas déraisonnable, et est probablement dans un facteur deux avec les coûts d'aujourd'hui. En clair, cela semble indiquer que le matériel de cassage de DES est à la portée d'un groupe bien plus large qu'en 1998, et il est important de noter que ceci ne suppose aucune amélioration de concept ou d'algorithme depuis lors.

Pour voir cela sous un angle légèrement différent, considérons l'interprétation normale de la Loi de Moore pour de telles discussions. Plutôt que de considérer la réduction de coût pour la même capacité, considérons à la place des capacités accrues pour le même coût (c'est à dire, de doubler la densité de circuits tous les N mois). Choisissons encore N=18 ; notre capacité de casser DES (dans le pire délai par clé) pourrait être suppose avoir suivi approximativement cette courbe de performances sur les sept dernières années environ :

- 1998 : 9 jours
- mi 2000 : 4,5 jours
- début 2002 : 2,25 jours
- mi 2003 : 1,125 jour
- début 2006 : 0,5625 jour

C'est juste un peu plus qu'une demi journée dans le pire cas pour 2006, et moins de 7 heures en moyenne. Et cela, pour un investissement de moins de 250 000 $. Il est aussi très important de noter que nous parlons ici du plus mauvais cas et de la moyenne – parfois, les clés seront trouvées beaucoup plus vite. Par exemple, en utilisant une telle machine, 1/4 de toutes les clés DES possibles seront trouvées en 3,375 heures. 1/8 des clés seront trouvées en moins d'une heure et 42 minutes. Et ceci ne suppose aucune amélioration algorithmique. Et encore, ceci est une estimation, ce n'est pas une règle absolue, mais l'estimation est probablement proche de la réalité.

7.1 Les FPGA

Depuis la première apparition de l'appareil de l'EFF, les réseaux prédiffusés programmables (FPGA, *Field Programmable Gate Arrays*) sont devenus assez courants, et beaucoup moins coûteux qu'ils ne l'étaient en 1998. Ces appareils permettent une programmation de faible niveau logique, et sont fréquemment utilisés pour des préséries de nouveaux concepts de logique avant la création de processeurs plus chers pour le commerce (on les appelle aussi des ASIC, *Application Specific Integrated Circuit*, circuit intégré spécifique d'application). Ils sont aussi fréquemment utilisés à la place des ASIC du fait

de leur plus faible coût et/ou de leur souplesse. En fait, un certain nombre de systèmes incorporés qui mettent en œuvre la cryptographie ont employé des FPGA à cette fin.

Étant par nature généralisés, les FPGA sont bien sûr plus lents que les ASIC. Bien que la différence de vitesse varie sur la base de nombreux facteurs, il est raisonnable de dire, pour les besoins de cet exposé, que les mises en œuvre bien conçues de FPGA effectuent normalement les opérations cryptographiques à environ 1/4 de la vitesse d'un ASIC bien conçu effectuant la même opération, et parfois beaucoup plus lentement que cela. La signification de cette comparaison sera bientôt évidente.

Dans notre estimation de la Loi de Moore ci-dessus, on a noté que l'extrapolation du coût ne suppose aucune amélioration de conception ni d'algorithme depuis 1998. Elle implique aussi que nous parlons toujours d'attaque en force. À la section 4 ("Attaquer DES"), nous exposons plusieurs attaques cryptanalytiques, y compris une attaque qui utilise l'analyse cryptographique linéaire [MAT93]. Cette attaque était en général considérée comme impraticable, mais en 2002, un groupe de l'Université catholique de Louvain en Belgique a construit un casseur de DES fondé sur l'analyse cryptographique linéaire, qui, en employant un seul FPGA, retourne une clé DES en 12 à 15 heures [FPL02].

Bien qu'il y ait toujours quelques questions pratiques pour l'application de cette attaque dans la réalité (en particulier le nombre de paires de texte clair – texte chiffré connues nécessaires) cela donne une idée de là où la technologie nous emmène par rapport aux capacités d'attaque de DES.

7.2 Les ASIC

Les circuits intégrés spécifiques d'application (ASIC, *Application Specific Integrated Circuit*) sont des processeurs spécialisés, normalement optimisés pour un ensemble particulier d'opérations (par exemple, le chiffrement). Un grand nombre d'entreprises sont spécialisées dans la conception et la vente d'ASIC cryptographiques, et de tels circuits intégrés peuvent être acquis pour 15 $ pièce au premier prix. Mais alors que ces circuits intégrés sont potentiellement bien plus rapides que les FPGA, ils ne représentent habituellement pas une menace proportionnellement plus forte quand elle vient de la construction de systèmes de cassage de DES.

La principale raison en est le coût : il coûte actuellement plus de 1 000 000 $ pour produire un ASIC. Il n'y a pas de grand marché commercial pour les ASIC de cassage de chiffre, aussi le nombre qu'un fabricant peut espérer en vendre est probablement faible. De même, un seul attaquant n'aura vraisemblablement besoin que de quelques uns d'entre eux. Au minimum, les coûts par puce seraient très élevés, comparés aux coûts de FPGA capables de performances similaires, les FPGA sont vainqueurs haut la main. Cela ne signifie pas que de tels ASIC n'ont jamais été construits, mais le retour sur investissement n'est pas suffisant pour l'attaquant moyen d'aujourd'hui, étant donné les autres options disponibles.

7.3 Les micro-ordinateurs distribués

Le traitement en parallèle est un outil puissant pour conduire des attaques en force contre un bloc de chiffrement. Comme chaque clé peut être testée indépendamment, l'espace de clé peut facilement être découpé et distribué à travers un nombre arbitraire de processeurs, fonctionnant tous avec un code identique. Un processeur central de "contrôle" est nécessaire pour les tâches de distribution et l'évaluation des résultats, mais c'est d'une mise en œuvre sans difficulté et ce paradigme a été appliqué à de nombreux problèmes d'informatique.

Alors que l'EFF a démontré qu'un système dédié est de loin supérieur à un micro-ordinateur tout venant pour casser DES, l'initiative DESCHALL [CURT05][RSA1] a parfaitement démontré que les cycles d'inactivité des micro-ordinateurs des utilisateurs ordinaires pouvaient être efficacement appliqués à ce problème. Comme noté ci-dessus, distributed.net en équipe avec le groupe EFF s'est appliqué à résoudre le troisième défi DES de RSA en utilisant une combinaison de micro-ordinateurs et la machine "Deep Crack" de l'EFF pour trouver une clé DES en 22 heures. Et cela, avec les technologies de 1999.

Il est clair que les micro-ordinateurs se sont considérablement améliorés depuis 1999. À cette époque, le micro-ordinateur moyen tournait environ à 800MHz. Aujourd'hui, les micro-ordinateurs tournent couramment trois à quatre fois plus vite, et prennent aussi en charge une offre technologique (mémoire, antémémoire, stockage) de performances bien plus élevées. Comme l'initiative distributed.net utilisait un large spectre d'ordinateurs (depuis les ordinateurs du début des années 1990 aux multiprocesseurs dernier cri (en 1999), selon [DIST99]), il est difficile de faire une comparaison directe avec les technologies d'aujourd'hui. Cependant, nous savons que les performances ont, en général, suivi la prédiction de la Loi de Moore, aussi devrait on s'attendre à une amélioration de l'ordre d'un facteur 8 à 16 actuellement, même sans amélioration algorithmique.

7.3.1 Participants volontaires

Il est important de noter que l'initiative distributed.net s'était appuyée sur des participants volontaires. C'est à dire que les participants doivent se joindre explicitement et de leur plein gré à l'initiative. Il est également important de noter que seuls

les cycles inactifs des systèmes participants sont utilisés. Selon la façon dont on définit "inactif", ainsi que selon les habitudes des usagers et les exigences de calcul, cela peut avoir un effet significatif sur le niveau de contribution d'un système donné.

Ces facteurs imposent des limitations significatives en termes d'échelle. Alors que distributed.net était capable d'enrôler plus de 100 000 ordinateurs du monde entier pour le troisième défi DES de RSA, cela ne représente en réalité qu'un nombre assez petit comparé aux 2^56 (plus de 72 quadrillions) de clés DES possibles. Et quand on considère le but (c'est à dire, de prouver que DES peut être cassé) il semble raisonnable de supposer que les mêmes participants n'offriraient pas volontiers leurs cycles de calcul pour un usage plus nocif (comme d'attaquer les clés utilisées pour chiffrer votre session de banque en ligne). Et donc, ce modèle particulier ne paraît pas poser une menace significative pour la plupart des utilisations de chiffrement d'aujourd'hui. Cependant, nous exposons ci-dessous une variante de cette approche qui fait bien peser une menace immédiate.

7.3.2 Les équipements espions, les virus et les réseaux zombies

Le logiciel espion *(spyware)* est actuellement un sujet populaire dans les magazines de sécurité. La plupart de ces applications sont destinées à afficher aux usager des publicités sensibles au contexte, et certaines à modifier en réalité ce que perçoit un utilisateur de navigateur sur l'Internet, le dirigeant sur des sites au choix du distributeur qui essaye d'en générer un revenu. Il y a de nombreux noms pour ce type de logiciel, mais pour ce qui nous concerne nous le désignerons simplement comme "logiciel espion". Et bien qu'il y ait un certain nombre d'instances dans lesquelles des logiciels escrocs espionnent effectivement d'infortunés usagers et en font rapport à leur auteur, nous ne sous attarderons pas ici à de telles distinctions.

En fait, nous sommes plus intéressés par les grandes lignes selon lesquelles fonctionne ce logiciel : il est normalement installé à l'insu de l'usager et sans qu'il le voit, et fonctionne normalement sans que l'usager le sache, ralentissant parfois le micro-ordinateur de l'usager à tel point qu'il donne l'impression de ramper. On pourra noter qu'un tel comportement paraît surprenant sachant que l'affichage de publicités aux utilisateurs est en réalité une tâche légère, et on doit alors se demander ce que ce logiciel fait de tous ces cycles de calcul.

Les vers et les virus sont aussi très intéressants : comme le logiciel espion, ils sont installés sans que l'utilisateur le sache ou y consente, et ils utilisent l'ordinateur d'une façon qu'il n'accepterait pas de son plein gré. À la différence du logiciel espion qui est le plus courant aujourd'hui, ce logiciel maléfique contient habituellement une technique explicite de propagation par laquelle il se répand automatiquement. Il n'est pas difficile d'imaginer où cela nous mène : si on combine ces techniques, la mobilisation forcée des machines des usagers en une "armée" de systèmes devient possible. Il était fait allusion à cette approche dans [CURT98] et en fait, c'est en train de se faire aujourd'hui.

Les réseaux zombies *(botnet)* [BOT05] représentent un phénomène relativement récent. En utilisant diverses techniques de propagation, le logiciel malveillant est distribué à travers toute une gamme de systèmes, où il est dormant en attendant un signal de déclenchement d'une sorte ou d'une autre. Ces "déclencheurs" peuvent être mis en œuvre par des interrogations périodiques par une autorité centralisée, l'arrivée d'une date particulière, ou n'importe lequel d'un grand nombre d'autres événements. Au signal, le logiciel malveillant exécute sa tâche, qui peut inclure la participation à une attaque distribuée de déni de service (DDoS), ou quelque autre type d'activité.

Des groupes criminels louent souvent des réseaux zombies pour divers usages [CERT01]. Tout en ayant des rapports d'occurrences ayant effectivement impliqué l'utilisation de ces réseaux malveillants pour des attaques de DDoS, on serait naïf de croire que d'autres usages (comme par exemple de casser des clés de chiffrement) n'ont pas été envisagés. Les réseaux zombies atténuent considérablement le problème d'échelle auquel était confronté distributed.net : ce n'est plus l'initiative de volontaires et l'activité de l'utilisateur n'entrave pas de façon significative le travail de l'application. Ceci devrait nous arrêter un instant.

Il est très important de reconnaître clairement les implications de cela : les réseaux zombies sont bon marché, et il y a des quantité de micro-ordinateurs. Vous n'avez pas besoin des 15 625 $ que nos spéculations estimaient nécessaires pour construire une copie d'aujourd'hui du système d'EFF – vous avez seulement besoin d'un micro-ordinateur sur lequel développer le logiciel malveillant, et les talents requis. Ou vous pouvez avoir accès à quelqu'un qui a cela, et pour une somme d'argent relativement modeste. Le jeu a dramatiquement changé.

8 Pourquoi DES est-il toujours utilisé ?

Selon toutes les mesures, visiblement, DES n'est pas sûre – pourquoi est-t-elle encore utilisée aujourd'hui ? Il peut y avoir à cela de nombreuses raisons, mais voilà peut-être les plus courantes :

- La rétrocompatibilité – De nombreux systèmes développés prennent en charge DES, et plutôt que de remplacer ces systèmes, les nouveaux systèmes sont mis en œuvre avec le souci de la compatibilité.
- Performance – De nombreux clients des premiers VPN fournissaient DES comme algorithme cryptographique par défaut, parce que les micro ordinateurs de l'époque souffraient d'un défaut de performances perceptible lorsqu'on

appliquait une cryptographie plus forte (par exemple, 3DES).

- Ignorance – Les gens ne comprennent tout simplement pas que DES n'est plus sûre pour la plupart des usages.

Bien qu'il y ait probablement d'autres raisons, celles-ci sont les plus fréquemment citées.

Les arguments sur les performances sont facilement avancés aujourd'hui. Les micro ordinateurs ont plus que largement la puissance pour mettre en œuvre une cryptographie plus forte sans impact perceptible sur les performances, et pour les systèmes qui ont des contraintes de ressources, il y a de forts algorithmes qui ont de bien meilleures performances que DES (par exemple, AES-128). Et bien que la rétrocompatibilité soit parfois un argument valable, cela doit être examiné attentivement. Lorsque le risque est plus fort que le coût de remplacement, les vieux systèmes devraient être abandonnés.

Pour ce qui concerne la troisième raison (l'ignorance), la présente note essaye de la résoudre, et nous devrions continuer à faire tous nos efforts pour y remédier. DES n'est plus sûre pour la plupart des utilisations, et il faut de bons experts en sécurité pour évaluer les quelques cas dans lesquels elle serait acceptable. Les technologies existent qui mettent la capacité de casser DES à la portée d'un attaquant motivé aux finances et à l'habileté modestes. Il y a des algorithmes de chiffrement plus forts, moins chers, plus rapides et disponibles. Il est temps de bouger.

9 Considérations sur la sécurité

Tout ce document traite de considérations de sécurité. Il paraît temps de résumer ici quelques points clés. Il devrait apparaître clairement maintenant que l'algorithme DES n'offre qu'une dissuasion illusoire contre un adversaire déterminé. Bien qu'il ait pu en coûter environ 250 000 $ pour construire un casseur de DES dédié en 1998, aujourd'hui, cela peut être fait pour considérablement moins. Bien sûr, les réseaux zombies sont d'une certaine façon gratuits, si on ne compte pas le temps de l'auteur du logiciel malveillant dans les coûts de calcul.

Est-ce que cela veut dire que DES ne devrait jamais être utilisée ? Et bien, non – mais cela veut bien dire que si on l'utilise cala doit être avec une extrême attention. Il est important d'évaluer soigneusement la valeur des informations protégées, à la fois pour leur propriétaire et pour un attaquant, et de bien cerner les risques potentiels. Dans certains cas, DES peut encore fournir un niveau de sécurité acceptable, par exemple, si vous voulez crypter un fichier sur le micro ordinateur familial, et qu'il n'y a pas de menaces réelle dans votre maisonnée.

Cependant, il est important de reconnaître que, dans un tel cas, DES est un peu comme le cadenas d'une armoire : il aide normalement les honnêtes gens à rester honnêtes, mais il n'arrêterait pas un voleur déterminé. Cela étant, il existe de forts algorithmes cryptographiques plus efficaces (par exemple, AES) ; il semble que la seule raison de continuer à utiliser DES aujourd'hui est une rétro compatibilité compulsive. Dans un tel cas, si il n'y a pas de plan pour éliminer graduellement de tels produits, vous pouvez, en tant que chargé de la sécurité, prendre les mesures suivantes :

- Recommander une approche de mise à niveau par étapes.
- Si possible, utiliser 3DES plutôt que DES (et dans tous les cas, NE PAS faire de DES l'algorithme par défaut !).
- Remplacer les clés avant de dépasser 2^32 blocs par clé (pour éviter diverses attaques d'analyse cryptographique).
- Si il y a une interface d'utilisateur, avertir les utilisateurs du fait que la cryptographie utilisée n'est pas forte, et pour votre application particulière, faire les recommandations appropriées à cet égard.

Le minimum : il est plus simple de ne pas utiliser cet algorithme que d'arriver aux quelques scénarios dans lesquels il peut l'être sans risque. Si vous avez de vieux systèmes qui s'appuient sur DES, il paraît sensé de commencer à les éliminer aussitôt que possible.

10 Remerciements

L'auteur remercie chaleureusement Doug Whiting, Matt Curtin, Eric Rescorla, Bob Baldwin, et Yoav Nir de leurs contributions. Leurs révisions, commentaires, et avis ont considérablement amélioré cette note. Et bien sûr, il nous faut remercier tout l'EFF et ceux qui se sont impliqués dans l'initiative "Deep Crack", DESCHALL, et distributed.net pour leurs recherches et réalisations de pionniers dans ce domaine.

Appendice A Qu'en est-il de 3DES ?

Il semble raisonnable, étant donné que nous recommandons d'éviter DES, de demander : qu'en est-il de 3DES ? Est elle sûre ? Heureusement, la plus grande partie de l'exposé ci-dessus ne s'applique pas à 3DES, et elle est en général encore "sûre". Nous expliquons brièvement ci-dessous pourquoi ceci est vrai, et quels risques existent actuellement.

A.1 Attaques en force contre 3DES

Rappelez vous qu'il y a pour DES 2^56 clés possibles, et qu'une attaque en force consiste à essayer chaque clé jusqu'à trouver la bonne. Comme les probabilités de trouver la clé au premier, second, ou même dernier essai, sont égales, nous nous attendons en moyenne à trouver la clé après la moitié des clés (2^55), ou après 36 028 797 018 963 968 déchiffrements. Cela ne semble pas complètement impossible étant donné les vitesses des processeurs actuels, et comme nous l'avons vu plus haut, on peut s'attendre avec la technologie d'aujourd'hui qu'une telle attaque pourrait presque certainement être menée à bien en environ douze heures.

Pour une attaque en force contre 3DES, la perspective est cependant beaucoup moins optimiste. Considérons le problème : nous connaissons C (et peut-être p), et nous essayons de deviner k1, k2, et k3 dans la relation suivante :

$$C = E\_k3(D\_k2(E\_k1(p)))$$

Afin de deviner toutes les clés, nous devons exécuter quelque chose comme ce qui suit (en supposant que k1, k2, et k3 sont des valeurs de 64 bits, comme Ci et p) :

```
pour ( k3 = 0 à 2^56 étape 1 )
    calculer C2 = D_k3(C1)
    pour ( k2 = 0 à 2^56 étape 1 )
        calculer C3 = E_k2(C2)
        pour ( k1 = 0 à 2^56 étape 1 )
          commencer
                calculer p = D_k1(C3) xor IV
                si ( p égal p-attendu )
                    sortie de boucle; les clés sont trouvées
          fin
```

Noter que dans le pire cas, la combinaison de clé correcte sera la dernière essayée, ce qui veut dire que nous avons essayé 2^168 opérations de chiffrement. Si on suppose que chaque déchiffrement 3DES (deux déchiffrements plus un chiffrement) prend une seule microseconde, cela nous fait un total de 1,19 x 10^37 ans. C'est bien plus que ce que les scientifique estiment pour la durée ce vie de l'univers.

Bien qu'il soit important de noter qu'on peut légèrement élaguer l'espace de clés en supposant que deux clés égales ne seront jamais utilisées (c'est-à-dire, k1 != k2, k2 != k3, k1 != k3), il n'en résulte pas une réduction significative du travail quand on considère la magnitude des nombres que nous avons à traiter. Et qu'en est-il si nous supposons que les avancées technologiques nous permettent de traiter DES beaucoup plus vite ?

Aujourd'hui le processeur commercial 3DES capable de chiffrement à 10 Gbit/s est largement disponible, et il traduit 15 625 000 de blocs DES par seconde. L'estimation donnée ci-dessus suppose 1 000 000 de blocs DES par seconde, donc le matériel à 10 Gbit/s est 15 fois plus rapide. Cela signifie que dans le pire cas cela prendrait 7,6 x 10^35 ans – pas beaucoup plus vite que dans le schéma plus grand.

Même si on considère un matériel qui serait un million de fois plus rapide, cela prendrait encore 7,6 x 10^29 ans – toujours plus que l'univers qui nous entoure. Visiblement, il n'y a rien à tirer de la vitesse. 3DES est, d'un point de vue pratique, probablement sûr contre les attaques en force pour le futur prévisible.

A.2 Attaques de cryptanalyse contre 3DES

À la différence de DES, il n'y a que peu d'attaques d'analyse cryptographiques connues contre 3DES. Ci-dessous, nous décrivons ces attaques qui sont actuellement exposées dans la littérature.

A.2.1 Attaques de rencontre en chemin

L'attaque la plus couramment décrite contre 3DES est la rencontre en chemin (MITM, *Meet-In-The-Middle*), décrite dans [HAC] et ailleurs. Elle fonctionne de la façon suivante : prendre une valeur de texte chiffré 'C' (avec la valeur de texte en clair correspondante 'p'), et calculer les valeurs de Cx = D_kx(C) pour toutes les (2^56) clés possibles. Mémoriser chaque paire Cx,kx dans un tableau indexé par Cx.

Calculons maintenant les valeurs de Cy = D_ki(E_kj(p)) dans une boucle imbriquée, comme illustré plus haut dans notre exercice de force brute. Pour chaque Cy, faites une boucle sur le tableau des Cx. Pour chaque correspondance trouvée, essayez le triple des clés. Il est important de noter qu'une correspondance n'implique pas que vous ayez les bonnes clés – vous devez essayer contre des paires supplémentaires de texte chiffré/texte en clair pour être certain (~3 paires pour une forte mesure de certitude avec 3DES). Finalement, il y aura exactement un triplet de clés correct.

Noter que le calcul du tableau initial des paires de Cx,kx exige 2^56 chiffrements et 2^56 blocs de mémorisation (environ 576 gigaoctets). Calculer les éléments d'exploration exige au plus 2^112 opérations cryptographiques (les explorations du tableau sont négligeables en comparaison), et 2^111 d'opérations en moyenne. Lucks [LUCKS] a fait des optimisations qui réduisent cela à environ 2^108.

3DES, même à la puissance de 2^108, est encore très fort. Si on utilise les limites de force brute précédentes (15 625 000 de blocs par seconde), cette attaque prendra de l'ordre de 6,586 x 10^17 ans pour se faire. Rendez la machine 1 million de fois plus rapide, et vous aurez toujours besoin de 658 MILLIARDS d'années. Vous êtes probablement en sécurité contre les attaques MITM sur 3DES pour l'avenir prévisible.

A.2.2 Attaques de clés en relations

Pour une description détaillée des attaques de clés en relation contre 3DES (et les autres algorithmes), voir [KELSEY]. En très bref, dans cette approche l'attaquant connaît le chiffrement d'un texte en clair donné sous la clé d'origine K, et quelques clés K'_i en relation avec elle. Il y a des attaques où l'attaquant choisit comment la clé va changer, et des attaques dans lesquelles la différence est connue, mais non contrôlée, par l'attaquant.

Voici comment cela fonctionne. Supposons la relation cryptographique suivante :

C = E_k3(D_k2(E_k1(p)))

La relation de clé est définie ci-après :

K = (k1,k2,k3) et K' = (k1 + d,k2,k3)

d étant une constante fixe. Connaissant p et C, on a besoin de décrypter C sous K' comme suit :

Soit kx = k1 + d (note : '+' représente xor)
et
p' = D_kx(E_k1(p))

Une fois que nous avons p', nous pouvons trouver kx en essayant de façon exhaustive chaque clé jusqu'à trouver une correspondance (2^56 chiffrements, dans le pire des cas). kx une fois trouvé, nous pouvons effectuer une attaque MITM de double-DES pour trouver k2 et k3, ce qui exige entre 2^56 et 2^72 d'essais de chiffrement hors ligne.

D'un point de vue pratique, il est très important de reconnaître la nature "spéculative" de cette attaque : l'adversaire doit connaître la paire texte en clair/texte chiffré, il doit être capable d'influencer une clé de chiffrement ultérieure d'une façon très contrôlée (ou au moins de savoir exactement comment change la clé), et puis avoir la coopération cryptographique adéquate pour calculer p'. Ceci est vraiment une attaque très difficile dans la réalité.

A.3 Taille de bloc 3DES

Bien que la taille effective de clé de 3DES soit clairement beaucoup plus grande que pour DES, la taille de bloc n'est, malheureusement, toujours que de 64 bits. Pour le mode CBC (le plus couramment déployé dans les protocoles de sécurité de l'Internet), cela signifie que, du fait du paradoxe de l'anniversaire, les informations sur le texte en clair commencent à se dévoiler après le déchiffrement d'environ 2^32 blocs. Pour cette raison, 3DES pourrait n'être pas le meilleur choix pour les liaisons à gros débit, ou autres applications de chiffrement à forte densité. Au minimum, il faut veiller à rafraîchir les clés suffisamment fréquemment pour minimiser les collisions de texte chiffré dans de tels scénarios.

Références informatives

[AES] "The Advanced Encryption Standard", novembre 2001, à
<http://csrc.nist.gov/publications/fips/fips197/ fips-197.pdf>.

[AES-NSA] "CNSS Policy No. 15, Fact Sheet No. 1", juin 2003, <http://csrc.nist.gov/cryptval/CNSS15FS.pdf>.

[BIH93] Biham, E. and A. Shamir, "Differential Cryptanalysis of the Data Encryption Standard", 1993.

[BIH96] Biham, E., "How to Forge DES-Encrypted Messages in 2^28 Steps", 1996.

[BIH96-2]Biham, E. ET A. Shamir, "A New Cryptanalytic Attack on DES", 1996.

[BLAZ96] Blaze, M., Diffie, W., Rivest, R., Schneier, B., Shimomura, T., Thompson, E., et M. Wiener, "Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security", janvier 1996.

[BOT05] "Know Your Enemy: Tracking Botnets", mars 2005, <http://www.honeynet.org/papers/bots/>.

[CERT01] Ianelli, N. et A. Hackworth, "Botnets as a Vehicle for Online Crime", décembre 2005, <http://www.cert.org/archive/pdb/Botnets.pdf>.

[CURT05] Curtin, M., "Brute Force: Cracking the Data Encryption Standard", 2005.

[CURT98] Curtin, M. and J. Dolske, "A Brute Force Search of DES Keyspace", 1998, <http://www.interhack.net/pubs/des-key-crack/>.

[DES] "Data Encryption Standard", janvier 1977, <http://www.nist.gov>.

[DH77] Hellman, M. et W. Diffie, "Exhaustive Cryptanalysis of the NBS Data Encryption Standard", juin 1977.

[DIST99] Communiqué de presse, distribué, "US GOVERNMENT'S ENCRYPTION STANDARD BROKEN IN LESS THAN A DAY", 1999, <http://www1.distributed.net/des/release-desiii.txt>.

[EFF98] EFF, "Cracking DES", juillet 1998.

[FBI01] "NIPC Advisory 01-003", mars 2001,<http://www.fbi.gov/pressrel/pressrel01/nipc030801.htm>.

[FBI06] "FBI Webpage: Focus on Economic Espionage", janvier 2006, <http://www.fbi.gov/hq/ci/economic.htm>.

[FERG03]Ferguson, N. and B. Schneier, "Practical Cryptography", 2003.

[FPL02] Koeune, F., Rouvroy, G., Standaert, F., Quisquater, J., David, J., et J. Legat, "An FPGA Implementation of the Linear Cryptanalysis", FPL 2002, Volume 2438 of Lecture Notes in Computer Science, pages 846-852, Spriger-Verlag, septembre 2002.

[HAC] Menezes, A., van Oorschot, P., et S. Vanstone, "Handbook of Applied Cryptography", 1997.

[JAK97] Jakobsen, T. et L. Knudsen, "The Interpolation Attack on Block Ciphers", 1997.

[KELSEY] Kelsey, J., Schneier, B., et D. Wagner, "Key-Schedule Cryptanalysis of 3-WAY, IDEA, G-DES, RC4, SAFER, and Triple-DES", 1996.

[LUCKS] Lucks, S., "Attacking Triple Encryption", 1998.

[MAT93] Matsui, M., "Linear Cryptanalysis Method for DES Cipher", 1993.

[NIST-TP] "DES Transition Plan", mai 2005, <http://csrc.nist.gov/cryptval/DESTranPlan.pdf>.

NYSE1] "Extreme availability: New York Stock Exchange's new IT infrastructure puts hand-held wireless terminals in brokers' hands.", juin 2005.

[RSA1] Communiqué de presse, RSA., "Team of Universities, Companies and Individual Computer Users Linked Over the Internet Crack RSA's 56-Bit DES Challenge", 1997, <http://www.rsasecurity.com/press_release.asp?doc_id=661&id=1034>.

[RSA2] Press Release, RSA., "RSA to Launch "DES Challenge II" at Data Security Conference", 1998, <http://www.rsasecurity.com/press_release.asp?doc_id=729&id=1034>.

[RSA3] Communiqué de presse, RSA., "Distributed Team Collaborates to Solve Secret-Key Challenge", 1998, <http://www.rsasecurity.com/press_release.asp?doc_id=558&id=1034>.

[RSA4] Communiqué de presse, RSA., "RSA to Launch DES Challenge III Contest at 1999 Data Security Conference", 1998, <http://www.rsasecurity.com/press_release.asp?doc_id=627&id=1034>.

[RSA5] Communiqué de presse, RSA., "RSA Code-Breaking Contest Again Won by Distributed.Net and Electronic Frontier Foundation", 1999, <http://www.rsasecurity.com/press_release.asp?doc_id=462&id=1034>.

[SCHN96] Schneier, B., "Applied Cryptography, Second Ed.", 1996.

[VA1] "Review of Issues Related to the Loss of VA Information Involving the Identities of Millions of Veterans (Report #06-02238-163)", July 2006, <http://www.va.gov/oig/51/FY2006rpts/VAOIG-06-02238-163.pdf>.

[WIEN94] Wiener, M., "Efficient DES Key Search", août 1993.

Adresse de l'auteur

Scott G. Kelly
Aruba Networks
1322 Crossman Ave
Sunnyvale, CA 94089
US
EMail: scott@hyperthought.com

Déclaration de copyright

Propriété intellectuelle

Remerciement

Le financement de la fonction d'édition des RFC est fourni par la Administrative Support Activity (IASA) de l'IETF.