TD Cryptographie

Jean-René Reinhard
jean-rene.reinhard@m4x.org

Mars 2016

Problème A : Utilisation de RC4 dans TLS

RC4, malgré ses nombreux défauts, est encore très utilisé de nos jours. Il est notamment mis en œuvre dans le protocole TLS, qui permet de chiffrer les données échangées dans une connexion TCP. Le protocole TLS fonctionne en deux étapes :

- dans un premier temps les interlocuteurs établissent de manière sécurisée des clés communes valides pour la durée de la session ;
- dans un deuxième temps ces clés sont utilisées pour protéger les communications, notamment les chiffrer.

Lorsque RC4 est mis en œuvre, il utilise des clés de 128 bits. Son initialisation ne fait pas appel à un vecteur d'initialisation. Pour chiffrer le flux de paquets TCP échangés, on utilise l'unique suite chiffrante produite à partir de la clé de session. Par exemple si le premier paquet compte 1500 octets, les 1500 premiers octets de la suite chiffrante sont utilisés pour le chiffrer. Si le deuxième paquet compte 100 octets, les 100 octets suivants de la suite chiffrante sont utilisés, ... Nous avons vu en cours que le WEP utilise un IV pour chiffrer chaque trame WiFi échangée.

1. Pourquoi la solution adoptée par TLS n'est pas envisageable pour le WiFi ? Autre formulation de cette même question : quelle est la propriété du protocole TCP qui permet de se passer d'IV dans le chiffrement ?

Des chercheurs ont récemment déterminé de manière expérimentale la distribution de probabilité des premiers octets de suite chiffrante de RC4, c'est à dire pour chaque position i de suite chiffrante plus petite qu'une certaine borne i_{limit}, pour chaque valeur d'octet z, déterminer la probabilité que $z_i = z$.

2. Écrire un algorithme permettant de déterminer de manière expérimentale la distribution du premier octet de suite chiffrante.

On donne en figure 1 la distribution de probabilité obtenue pour le deuxième octet de suite chiffrante : pour chaque valeur en abscisse, on lit en ordonnée la probabilité d'appartion de la valeur.

3. Quelle distribution attend-t-on pour un bon GPA ? Quel défaut observe-t-on ?

On peut exploiter les défauts indentifiés à la question précédente pour mettre en place une attaque dans un scénario multi-sessions. On suppose que l'attaquant peut obtenir le chiffré d'un même message clair sous D clés différentes. Ce scénario correspond à une réalité : les premiers messages de certains protocoles envoient de manière systématique un mot de passe.

4. Décrire une attaque permettant d'exploiter les D chiffrés d'un même message clair permettant de retrouver le deuxième octet du message clair.

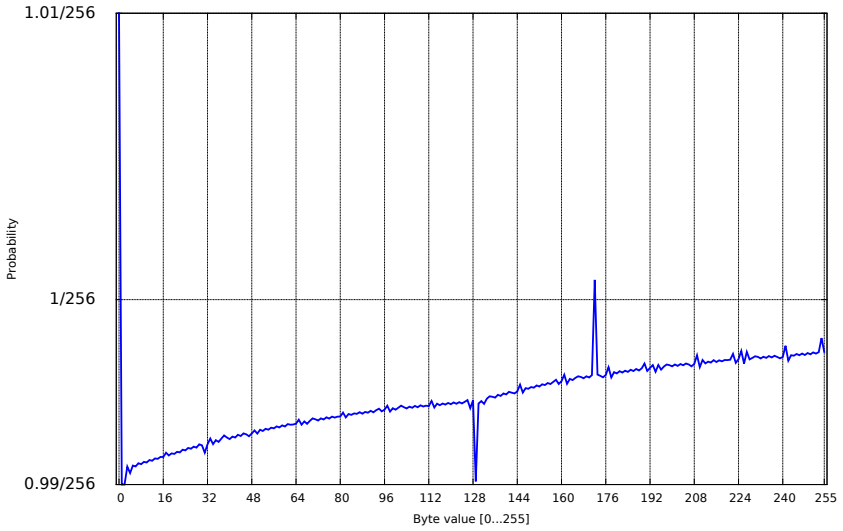


FIGURE 1 – Distribution de probabilité du deuxième octet de sortie de RC4

En pratique l'attaque décrite à la section précédente permet de retrouver le deuxième octet de clair avec de l'ordre de $D = 2^{20}$ chiffrés. Pourtant l'attaque reste difficilement applicable en pratique.

5. À votre avis, quel facteur limite l'impact pratique de l'attaque décrite ?

Problème B : Passeport électronique

Les passeports électroniques comportent en bas de la page d'identification deux bandes de caractères. Cette zone peut-être lue par un terminal de manière optique. La deuxième ligne comporte un numéro d'identification, appelé MRZ (Machine Readable Zone), qui fait office de clé pour protéger les communications électroniques entre le passeport et le terminal, dans le protocole BAC. L'objectif est d'assurer la confidentialité des échanges wireless entre un passeport et un terminal de lecture contre un attaquant ne pouvant pas un accès visuel sur le passeport. Outre des sommes de contrôle et du bourrage, cette MRZ est constituée de :

- un numéro d'identification composé de 9 caractères, lettre en majuscules ou chiffre, ou uniquement chiffre ;
- la date de naissance du porteur du passeport, au format YYMMJJ ;
- la date d'expiration du passeport, au format YYMMJJ ;
- le sexe du porteur, M, F ou < (non spécifié)

1. Quel est le nombre de valeurs possibles pour la MRZ d'un passeport ?

2. On observe une personne présentant un passeport à la douane, sans pouvoir lire la MRZ. Comment le nombre de MRZ possibles est-il réduit ?

Le protocole BAC, pour Basic Access Control, suit le déroulement suivant :

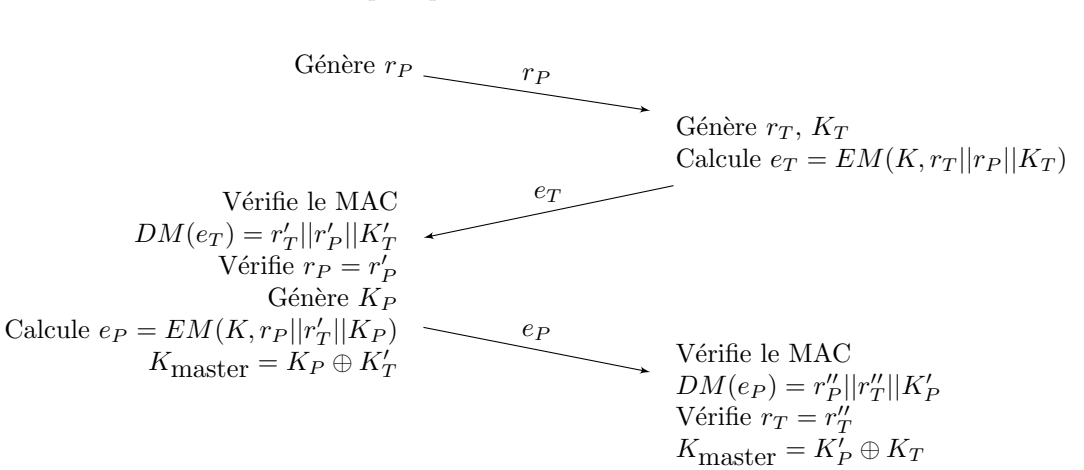
- Le passeport dispose d'une clé de chiffrement K_E et d'une clé de MAC K_M, on note $K = (K_E, K_M)$.
- Le terminal lit la MRZ et en déduit K.

– Les échanges font appel à une fonction de chiffrement intègre

$$EM(K, S) = E(K_E, S)||MAC(K_M, E(K_E, S)).$$

La fonction de déchiffrement intègre correspondante est notée DM.

– Déroulement du protocole



3. Montrer qu'un attaquant passif, c'est à dire qui se contente d'écouter une exécution complète du protocole, peut en déduire un test d'arrêt, permettant de réaliser une recherche exhaustive sur K_E.

4. En considérant les deux premières questions, que peut-on dire de la protection en confidentialité apportée par le BAC ?

Problème C : Propriétés de RSA

RSA est un schéma asymétrique pouvant être utilisé aussi bien en chiffrement qu'en déchiffrement. La clé publique est constitué d'un module RSA N, produit de deux grands facteurs premiers p et q gardés secret, et d'un entier e appelé exposant public. La clé privée est constituée d'un exposant privée d, inverse de e modulo $\varphi(N) = (p-1) \cdot (q-1)$, i.e., $ed = 1 \mod \varphi(N)$.

1. Montrer qu'on peut déduire d de la clé publique et de la factorisation de N.

On s'intéresse désormais à la propriété réciproque. On suppose que l'adversaire connait d.

2. Montrer qu'il peut en déduire un multiple λ de $\varphi(N)$.

3. Justifier que λ est pair.

Par la suite on note $\lambda = 2^r\gamma$, avec γ impair. On rappelle également que 1 possède 4 racines distinctes dans $\mathbb{Z}/N\mathbb{Z}$, qui correspondent, par le théorème des restes chinois, aux 4 paires suivantes de $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$:

$$\begin{array}{rrrrrcrr}
(& 1 & \mod p, & 1 & \mod q) & = & 1 & \mod N \\
(& -1 & \mod p, & -1 & \mod q) & = & -1 & \mod N \\
(& 1 & \mod p, & -1 & \mod q) & = & \alpha & \mod N \\
(& -1 & \mod p, & 1 & \mod q) & = & -\alpha & \mod N
\end{array}$$

4. Soit g un élément de $(\mathbb{Z}/N\mathbb{Z})^*$. Quelle est la valeur de g^λ ?

5. Justifier qu'on peut considérer $g^{\lambda/2}$.

6. Quelles sont les valeurs possibles pour $g^{\lambda/2}$?

7. Supposons que $g^{\lambda/2} = \alpha$. Calculer $g^{\lambda/2} - 1$ modulo p et modulo q. En déduire la factorisation de N.

8. Que peut-on faire dans le cas où $g^{\lambda/2} = 1$? dans le cas où $g^{\lambda/2} = -1$?

9. Écrire un algorithme probabiliste permettant de factoriser N à partir de e et d.

On suppose que deux utilisateurs possèdent des bi-clés RSA partageant le même module : (N, e_1) et (N, e_2)

10. Que peut-on dire de la confidentialité des messages adressés au premier utilisateur ?

11. Le même message m est chiffré à destination des deux utilisateurs. Montrer que le message m peut être récupéré à partir des chiffrés c_1 et c_2, sans utiliser les exposants privés (Indication : considérer une identité de Bezout).

En pratique e est choisi petit.

12. Quel avantage y a-t-il à choisir e petit ?

13. Notons N_1, N_2 et N_3 les modules RSA de trois utilisateurs, utilisant pour exposant public $e = 3$. On chiffre un même message m à destination de ces trois destinataires, et on obtient les chiffrés c_1, c_2 et c_3. Montrer qu'un adversaire interceptant ces chiffrés peut retrouver m sans connaître les clés privées des destinataires.

Par conséquent on choisit des valeurs de e supérieures, mais restant petites. typiquement, $e = 2^{16} + 1$.

14. e étant choisi petit, pourquoi d ne peut-il pas être également petit ?

Afin d'accélérer les calculs en déchiffrement (ou en signature), on considère la stratégie suivante : on réalise les calculs modulo p et modulo q, puis on utilise le théorème des restes chinois pour combiner les résultats.

15. Écrire les opérations réalisées pendant le déchiffrement

16. Quel est le gain en temps obtenu ?

17. On se place désormais dans le scénario d'une attaque par faute pendant une signature. On suppose que la faute produit une erreur uniquement pendant le calcul réalisé modulo p. Montrer qu'on peut obtenir la factorisation de N à partir de la signature erronée.

18. Proposer une contremesure contre cette attaque.

Problème D : Sécurité du $2DES$

L'algorithme de chiffrement DES souffre d'une taille de clé insuffisante (56 bits) et d'une taille de bloc un peu courte (64 bits). Afin de pouvoir utiliser des tailles de clés plus importantes, des constructions du DES en cascade ont été proposées. Elles consistent à composer plusieurs appels à l'algorithme de chiffrement par bloc DES, appels utilisant des clés différentes. La taille de bloc est inchangée.

$$\begin{aligned} 2DES_{K_1,K_2}(P) &= DES_{K_2}(DES_{K_1}(P)), \\ 3DES_{K_1,K_2}(P) &= DES_{K_1}(DES^{-1}_{K_2}(DES_{K_1}(P))), \\ 3DES_{K_1,K_2,K_3}(P) &= DES_{K_3}(DES^{-1}_{K_2}(DES_{K_1}(P))), \end{aligned}$$

1. Pourquoi une taille de bloc de 64 bits est-elle insuffisante en règle générale de nos jours ?

2. Donner les tailles de clé pour chacune des 3 constructions présentées ci-dessus.

On s'intéresse à la sécurité de $2DES$. On suppose disposer de quelques paires (clair, chiffré). On considère en particulier (P, C). On construit la table T_1 obtenue en considérant l'ensemble des chiffrés de P sous toutes les clés K_1^* possibles : $T_1 = \{DES_{K_1^*}(P)\}$.

3. Que peut-on dire du déchiffré de C sous K_2 ?

4. En déduire une attaque contre le $2DES$. On prendra soin d'éviter les fausses alarmes. Donner sa complexité en temps et en mémoire. Comparer à la recherche exhaustive naïve.

Problème E : MAC

On s'intéresse dans ce problème à des schémas d'authentification de message symétriques basés sur un algorithme de chiffrement par bloc E, dont la taille de clé est notée k et la taille de bloc est notée n. Pour simplifier, on ne considère que des messages M constitués d'un nombre entier de blocs : $M = M_1||\dots||M_\ell$.

CBC-MAC calcule un MAC de M à partir d'une chaîne CBC de M.

$$\begin{aligned} X_0 &= 0 \\ X_i &= E_K(X_{i-1} \oplus M_i), 1 \leq i \leq \ell \\ \text{CBC-MAC}_K(M) &= X_\ell \end{aligned}$$

1. Faire un schéma du fonctionnement de cet algorithme de génération de MAC.

2. Montrer qu'on peut, à partir des MAC, μ_0, μ_1 de deux messages M^0, M^1 réaliser une forge pour un troisième message. *Indication :* considérer la concaténation $M_0||M_1$ et adapter ce qui doit l'être.

3. Quelle limitation peut-on imposer à la taille des messages pour éviter cette attaque ?

Pour éviter les problèmes identifiés dans la question précédente, une solution est de réaliser un surchiffrement en bout de chaîne CBC. On dénote ce schéma EMAC. Il utilise deux clés K_1 et K_2 indépendantes de taille k.

$$\begin{aligned} X_0 &= 0 \\ X_i &= E_{K_1}(X_{i-1} \oplus M_i), 1 \leq i \leq \ell \\ \text{EMAC}_{K_1,K_2}(M) &= E_{K_2}(X_\ell) \end{aligned}$$

4. Expliquer pourquoi le surchiffrement final évite le problème rencontré avec CBC-MAC.

Quelques problèmes subsistent néanmoins lorsque l'algorithme de chiffrement par bloc est mal dimensionnée, comme c'est le cas dans le DES-retail-MAC qui utilise le DES comme algorithme de chiffrement par bloc.

5. Que peut-on dire sur les MAC de deux messages M^0, M^1, pour lesquels on a $X^0_{\ell_0} = X^1_{\ell_1}$?

6. Proposer une attaque exploitant cette propriété. *Indication :* considérer que l'adversaire accède à un grand nombre de paires (message, MAC). Que doit-il faire pour trouver deux messages dont le calcul de MAC vérifie la propriété de la question précédente ? Comment l'attaquant peut-il retrouver K_1 à partir de ces deux messages ? et K_2 ?

7. Application numérique : quelle est la complexité (en temps, en données) de l'attaque précédente dans le cas du DES-retail-MAC ?

Problème F : Chiffrement El-Gamal

L'objectif de cet exercice est d'étudier la sécurité du schéma de chiffrement ElGamal. On rappelle brièvement le fonctionnement de ce système cryptographique asymétrique, qui repose sur le problème du logarithme discret :

- **Paramètres du schéma.** Soit q un nombre premier de grande taille et g un générateur du groupe multiplicatif $(\mathbb{Z}/q\mathbb{Z})$. L'algorithme de génération de clés tire aléatoirement et uniformément un élément x de $(\mathbb{Z}/q\mathbb{Z})$ et calcule $y = g^x \mod q$. La clé publique est la donnée (q, g, y) et la clé secrète associée est x.
- **Chiffrement.** Etant donné un message clair $m \in (\mathbb{Z}/q\mathbb{Z})$, l'algorithme de chiffrement tire aléatoirement et uniformément un élément $r \in (\mathbb{Z}/q\mathbb{Z})$ et calcule $c_1 = g^r \mod q$ et $c_2 = my^r \mod q$. Le chiffré de m est alors défini comme étant le couple $c = (c_1, c_2)$.

1. Expliquer comment fonctionne le déchiffrement d'un chiffré $c = (c_1, c_2)$ en m. Quelle condition doit vérifier c_1 pour que m soit défini ? Montrer que cette condition est vérifiée quand le chiffré est calculé en suivant la description ci-dessus.

On souhaite montrer qu'il est possible d'inverser le schéma de chiffrement ElGamal, i.e. de retrouver la valeur de m à partir de c, dans le cadre d'une attaque à chiffré choisi.

2. Rappeler la notion d'"attaque à chiffré choisi".

3. On considère un message clair m et un chiffré $c = (c_1, c_2)$ de m par ElGamal. On définit à présent $c'_1 = (c_1 \cdot g^\alpha \mod q)$ et $c'_2 = (c_2 \cdot y^\alpha \mod q)$ pour $0 < \alpha < (q-1)$. Que représente le couple $c = (c_1, c_2)$ par rapport à m ?

4. Expliquer alors comment, en envoyant c' à un oracle de déchiffrement, il est possible de retrouver le déchiffrement de c ?

On souhaite maintenant renforcer la sécurité du schéma de chiffrement ElGamal en ajoutant un élément supplémentaire, noté c_3 , au chiffré. On supposera donc à présent que le chiffré d'un message m s'écrit sous la forme $c = (c_1, c_2, c_3)$ où c_1 et c_2 sont définis comme dans le schéma ElGamal initial et où c_3 est égal à $H(X)$, avec H une fonction de hachage et X un élément à déterminer. Nous appelons Hash-ElGamal ce nouveau schéma. Dans la suite de l'exercice, nous allons étudier comment le choix de X peut influencer la sécurité du système. Il est important de noter ici que lors du déchiffrement d'un chiffré c, on vérifiera d'abord que le haché en c_3 est correct, avant de retourner le message clair m. Dans la suite, on pose (c_1, c_2, c_3) le chiffré par Hash-ElGamal d'un message m dont un adversaire cherche à obtenir le déchiffrement.

5. Cas 1 : $X = g^r$. Trouver un deuxième chiffré c' de m. Conclure quant à la résistance aux attaques à chiffré choisi de cette proposition.

6. Cas 2 : $X = y^r$. Trouver un chiffré du message mg. Conclure quant à la résistance aux attaques à chiffré choisi de cette proposition.

7. Cas 3 : $X = (m, y^r)$. Expliquer intuitivement pourquoi ce cas semble empêcher d'attaquer Hash-ElGamal dans un modèle d'attaque à chiffré(s) choisi(s).

Problème G : MAC Parallélisable

Cette exercice montre les difficultés liés à la construction d'un schéma d'authentification de message parallélisable, i.e., capable de traiter les blocs de message indépendamment les uns des autres. Soit $M = (M_1, \ldots, M_m)$ un message formé de m blocs de n bits chacun. On suppose donc que le message a été complété par un padding adapté, de manière qu'il se compose d'un nombre entier de blocs. Dans tout l'exercice, on considère une fonction F se comportant de manière aléatoire, prenant en entrée une clé K de taille k bits et un bloc B de taille b bits et fournissant en sortie un bloc T de taille t bits.

1. Dans cette question, prenons $b = n$ et on considèrons la fonction de MAC

$$\mathrm{MAC}_K(M) = F_K(M_1) \oplus \ldots \oplus F_K(M_m).$$

   Montrer qu'avec cette définition il est facile, étant donné $(M, \mathrm{MAC}_K(M))$, de trouver un message $M' \neq M$ tel que $\mathrm{MAC}_K(M') = \mathrm{MAC}_K(M)$. Conclure quant à la sécurité de cette fonction de MAC.

Pour renforcer le schéma, on prend maintenant $b = n+64$ et on considère la nouvelle fonction MAC

$$\mathrm{MAC}_K(M) = F_K(<1>||M_1) \oplus \ldots \oplus F_K(<m>||Mm),$$

où $<i>$ est la représentation binaire de l'entier i sur 64 bits.

2. Montrer qu'il est possible de construire quatre messages P, Q, R, S tels que pour toute clé K on ait :

$$\mathrm{MAC}_K(P) \oplus \mathrm{MAC}_K(Q) \oplus \mathrm{MAC}_K(R) \oplus \mathrm{MAC}_K(S) = 0.$$

   En déduite un MAC valide pour S à partir de MAC pour les autres messages.

Remarques / Questions

N'hésitez pas à contacter l'auteur en cas de typos ou de questions.