Master CSI 2 2020-2021

Cryptanalyse — 4TCY902U
Responsable : G. Castagnos

Projet de cryptanalyse

Projet individuel. Vos programmes clairs et bien commentés ainsi qu'un rapport contenant les réponses aux questions (démonstrations, explications du code Sage) sont à rendre avant le mardi 8 décembre 23 :59 sur moodle. La clarté de la présentation de votre rapport et du code sera prise en compte dans la notation.

1 Cryptanalyse de A5/2

Cette première partie du projet consiste en une cryptanalyse de l'algorithme de chiffrement par flot A5/2 en s'inspirant de la méthode proposée par Barkan, Biham et Keller en 2003[1].

A5/2 est constitué de 4 LFSR, notés LFSR_1, LFSR_2, LFSR_3 et LFSR_4, de longueurs respectives $19, 22, 23$ et 17 et de polynômes de rétroactions respectifs $P_1 = x^{19} + x^{18} + x^{17} + x^{14} + 1$, $P_2 = x^{22}+x^{21}+1$, $P_3 = x^{23}+x^{22}+x^{21}+x^{8}+1$ et $P_4 = x^{17}+x^{12}+1$. Il utilise une clef $K = (K_0, \dots, K_{63})$ de 64 bits et un $IV = (IV_0, \dots, IV_{21})$ de 22 bits et il produit une suite chiffrante z de longueur fixe $N = 228$ bits. On notera à chaque instant $R_1 = (R_{1,0}, \dots, R_{1,18}), R_2 = (R_{2,0}, \dots, R_{1,21}), R_3 = (R_{3,0}, \dots, R_{3,22})$ et $R_4 = (R_{4,0}, \dots, R_{4,16})$, les registres des 4 LFSR.

La phase d'initialisation, $\mathsf{A5/2-init}$, est décrite dans l'encadré suivant :

Mettre à zéro les registres R_1, R_2, R_3, R_4
Pour i allant de 0 à 63,
Mettre à jour les LFSR_1, LFSR_2, LFSR_3, LFSR_4 en ignorant leurs sorties
$R_{1,18} = R_{1,18} + K_i, R_{2,21} = R_{2,21} + K_i, R_{3,22} = R_{3,22} + K_i, R_{4,16} = R_{4,16} + K_i$
Pour i allant de 0 à 21,
Mettre à jour les LFSR_1, LFSR_2, LFSR_3, LFSR_4 en ignorant leurs sorties
$R_{1,18} = R_{1,18} + IV_i, R_{2,21} = R_{2,21} + IV_i, R_{3,22} = R_{3,22} + IV_i, R_{4,16} = R_{4,16} + IV_i$
On fixe à 1 certains bits des registres : $R_{1,3} = 1, R_{2,5} = 1, R_{3,4} = 1, R_{4,6} = 1$.

[1] *Instant Ciphertext-Only Cryptanalysis of GSM encrypted communication*, Elad Barkan, Eli Biham, Nathan Keller, CRYPTO 2003

On note $\mathsf{Maj}(a,b,c)$ la fonction majorité. On rappelle que $\mathsf{Maj}(a,b,c) = 0$ si et seulement si la majorité des bits a, b et c vaut 0, par exemple $\mathsf{Maj}(0,1,0) = 0$ et $\mathsf{Maj}(1,1,0) = 1$.

$\mathsf{A5/2}$ utilise la fonction de mise à jour $\mathsf{A5/2-step}$ décrite dans l'encadré suivant :

Calculer $m = \mathsf{Maj}(R_{4,6}, R_{4,13}, R_{4,9})$
Si $R_{4,6} = m$, le $LFSR_1$ est mis à jour en ignorant sa sortie
Si $R_{4,13} = m$, le $LFSR_2$ est mis à jour en ignorant sa sortie
Si $R_{4,9} = m$, le $LFSR_3$ est mis à jour en ignorant sa sortie
Le $LFSR_4$ est mis à jour en ignorant sa sortie
Calculer $y_1 = R_{1,0} + \mathsf{Maj}(R_{1,3}, R_{1,4} + 1, R_{1,6})$
Calculer $y_2 = R_{2,0} + \mathsf{Maj}(R_{2,8}, R_{2,5} + 1, R_{2,12})$
Calculer $y_3 = R_{3,0} + \mathsf{Maj}(R_{3,4}, R_{3,9} + 1, R_{3,6})$
Le bit de sortie de $\mathsf{A5/2-step}$ est $y_1 + y_2 + y_3$

Après l'exécution phase d'initialisation, $\mathsf{A5/2-init}$, la production de $N = 228$ bits de suite chiffrante, se fait de la manière suivante :

- Exécuter 99 fois la fonction $\mathsf{A5/2-step}$ en ignorant son bit de sortie
- Exécuter $N = 228$ fois la fonction $\mathsf{A5/2-step}$, en utilisant ses bits de sortie pour produire la suite chiffrante, z.

$\boxed{1}$ Programmer (avec Sage) le chiffrement $\mathsf{A5/2}$. Il sera utile pour la suite de diviser le code en plusieurs fonctions (initialisation, mise à jour de l'état, production de suite chiffrante). Pour vérifier votre code, on pourra trouver dans

`https://www.math.u-bordeaux.fr/~gcastagn/Cryptanalyse/A5_2-test-vector.sage`

un exemple de suite chiffrante créée par $\mathsf{A5/2}$.

$\boxed{2}$ On se place juste après la phase d'initialisation. On suppose connu le registre R_4 du $LFSR_4$. On pose $R_1 = (x_0, \dots, x_{18}), R_2 = (x_{19}, \dots, x_{40})$ et $R_3 = (x_{41}, \dots, x_{63})$ où les x_i sont des inconnues (sauf 3 x_i que l'on sait être égaux à 1). Montrer que durant toutes les étapes de la production de suite chiffrante de $\mathsf{A5/2}$, on peut exprimer les contenus des registres R_1, R_2, R_3 au moyen d'équations linéaires en les x_i.

$\boxed{3}$ Écrire un code Sage permettant d'exprimer ces équations linéaires. Pour cela, déclarer les inconnues par `BPR = BooleanPolynomialRing(64,'x'); v = BPR.gens()` et utiliser les matrices de rétroaction des LFSR.

$\boxed{4}$ Donner la forme algébrique normale de la fonction booléenne Maj. En déduire que les bits de suite chiffrante z peuvent s'exprimer par des équations quadratiques en les x_i.

$\boxed{5}$ Écrire un code Sage permettant d'exprimer ces N équations quadratiques.

$\boxed{6}$ Montrer qu'au plus 655 monômes de degré au plus deux peuvent apparaître dans ces équations (sachant que l'on connaît la valeur de trois x_i). Créer avec Sage la liste M de ces monômes. On note $L = 655$ la longueur de M.

$\boxed{7}$ Linéariser ces équations avec Sage. Pour cela, créer une matrice $N \times L$ à coefficients dans $\mathbf{F}_2$ dont la i–ième ligne contient un 1 à la colonne j si et seulement si le monôme M_j apparaît dans l'équation i. Créer également un vecteur de longueur N qui contient un 1 à la coordonnée i si un 1 apparaît dans l'équation i. On pourra utiliser la méthode `.monomials()` appliquée à une équation pour obtenir la liste des monômes qui la constitue.

$\boxed{8}$ On suppose dans les trois questions suivantes que $N = 700$. Récupérer dans le fichier

`https://www.math.u-bordeaux.fr/~gcastagn/Cryptanalyse/A5_2-700.sage`

une suite chiffrante de 700 bits créée par A5/2 et la valeur du registre R_4 à la fin de la phase d'initialisation. Quel était le contenu des registres R_1, R_2, R_3 ? Pour résoudre un système linéaire, on pourra utiliser la méthode `.solve_right()` appliquée à une matrice.

$\boxed{9}$ On cherche maintenant à retrouver la clef secrète. Montrer que lors de l'étape d'initialisation, après la première boucle for, l'état du $LFSR_1$ est égal à

$$X = \sum_{i=0}^{63} A^i{}_1 \begin{pmatrix} 0 \\ \vdots \\ 0 \\ K_{63-i} \end{pmatrix}$$

où A_1 est la matrice de rétroaction du $LFSR_1$. Montrer qu'après le deuxième boucle for, l'état du $LFSR_1$ est égal à

$$A_1^{22} X + \sum_{i=0}^{21} A^i{}_1 \begin{pmatrix} 0 \\ \vdots \\ 0 \\ IV_{21-i} \end{pmatrix}$$

$\boxed{10}$ L'IV utilisé pour créer la suite chiffrante de 700 bits était nul. Déduire de la question précédente, avec Sage, les équations exprimant les contenus des registres R_1, R_2, R_3 et R_4 en fin de phase d'initialisation en fonctions des bits $(K_0, \dots, K_{63})$. Quel était la clef secrète utilisée pour construire cette suite de 700 bits ?

On revient pour toute la suite au cas $N = 228$ bits. Récupérer dans le fichier

```
https://www.math.u-bordeaux.fr/~gcastagn/Cryptanalyse/A5_2-3frames.sage
```

3 suites chiffrantes z_0, z_1 et z_2 chacune de 228 bits et produites avec A5/2 initialisé avec la même clef K et respectivement avec les IV $(0, \dots, 0), (0, \dots, 0, 1)$ et $(0, \dots, 0, 1, 0)$ (Ce cas correspond à l'utilisation d'A5/2 dans le protocole GSM).

$\boxed{11}$ On se place après la phase d'initialisation de l'exécution de A5/2 ayant produit la suite z_0. Comme à la question 2, on suppose connu le registre R_4 du $LFSR_4$ et on pose $R_1 = (x_0, \dots, x_{18})$, $R_2 = (x_{19}, \dots, x_{40})$ et $R_3 = (x_{41}, \dots, x_{63})$ où les x_i sont des inconnues. Montrer comment déduire de R_1, R_2, R_3 et R_4, l'état des registres après la phase d'initialisation des exécutions de A5/2 ayant produit les suites z_1 et z_2.

$\boxed{12}$ Déduire de la question précédente et de ce qui a été fait précédemment pour le cas $N = 700$, une attaque permettant de retrouver la clef K utilisée.

$\boxed{13}$ Programmer cette attaque (la valeur du registre R_4 après la phase d'initialisation de l'exécution de A5/2 donnant z_0 est donnée dans le fichier contenant z_0, z_1 et z_2).

$\boxed{14}$ En déduire une attaque (théorique) permettant de retrouver la clef K sans connaître R_4. Vu le temps pris par l'attaque de la question 13, combien de temps prendrait l'exécution de cette attaque complète ?

$\boxed{15}$ Proposer des changements dans la conception d'A5/2 le mettant à l'abri de cette attaque.

2 Cryptanalyse de RSA

Cette deuxième partie consiste en une attaque sur RSA exploitant la réduction de réseaux euclidiens, due à Boneh et Durfee[2]. Cette attaque utilise des petites racines d'un polynôme à deux variables, étendant la méthode de Coppersmith vue en cours.

On note $N = pq$ un module RSA où p, q sont deux premiers distincts aléatoires de k bits. On désigne par e l'exposant public et d la clef privée telle que $ed \equiv 1 \pmod{\varphi(n)}$. On supposera dans toute la suite que $d \approx N^{\delta}$ avec $\delta \in \mathbf{R}$, $0 < \delta < 1$.

$\boxed{16}$ Implanter un algorithme de génération de clefs de RSA avec cette contrainte sur d : il prend k et δ en entrée et ressort N, e, d avec d la plus grande clef privée possible inférieure à N^{δ}.

Dans la suite, on pose x_0 l'entier tel que $ed = 1 + x_0\varphi(n)$ et $y_0 := -(p+q)$.

$\boxed{17}$ On note $A = N+1$. Montrer que (x_0, y_0) est une racine du polynôme à deux variables $f(x, y) := 1 + x(A + y)$ modulo e. On suppose que $e \approx \varphi(N) \approx N$. Montrer que $x_0 \approx e^{\delta}$ et $|y_0| \approx e^{1/2}$.

[2] *Cryptanalysis of RSA with Private Key d Less than* $N^{0.292}$, Dan Boneh, Glenn Durfee, EUROCRYPT 1999

On introduit une nouvelle variable u avec la relation $u = 1 + xy$ de telle sorte qu'en posant $\bar{f}(u,x) := u + \mathrm{A}x$, on ait $\bar{f}(u,x) = f(x,y)$.

Soit m et t deux entiers avec $m \geqslant 2$ et $m \geqslant t \geqslant 1$. On considère les familles de polynômes

$$\bar{g}_{i,k}(u,x) = x^i \bar{f}^k e^{m-k} \text{ pour } k \in \{0,\dots,m\} \text{ et } i \in \{0,\dots,m-k\}$$

et

$$\bar{h}_{j,k}(u,x,y) = y^j \bar{f}^k e^{m-k} \text{ pour } j \in \{1,\dots,t\} \text{ et } k \in \{\lfloor m/t \rfloor j,\dots,m\}$$

De plus on considère la famille de monômes,

$$\mathrm{M} := \left\{x^{k-i}u^i \text{ pour } k \in \{0,\dots,m\} \text{ et } i \in \{0,\dots k\}\right\} \cup \left\{y^j u^k \text{ pour } j \in \{1,\dots,t\} \text{ et } k \in \{\lfloor m/t \rfloor j,\dots,m\}\right\}$$

$\boxed{18}$ On note $u_0 = 1 + x_0 y_0$. Montrer que tout polynôme $\bar{g}_{i,k}$ (resp. $\bar{h}_{j,k}$) a la racine (u_0, x_0) (resp. (u_0, x_0, y_0)) modulo e^m.

$\boxed{19}$ On note ℓ le cardinal de M. Montrer qu'on a en tout ℓ polynômes $\bar{g}_{i,k}$ et $\bar{h}_{j,k}$.

Soit X, Y tels que $|x_0| \leqslant \mathrm{X}$ et $|y_0| \leqslant \mathrm{Y}$. On note de plus $\mathrm{U} = 1 + \mathrm{XY}$. Comme dans la méthode de Coppersmith originale vue en cours, on construit une matrice $\ell \times \ell$ base d'un réseau $\mathscr{L}$. Les vecteurs de bases sont constitués des vecteurs de coefficients des polynômes $\bar{g}_{i,k}(u\mathrm{U}, x\mathrm{X})$ et des vecteurs de coefficients des polynômes $\bar{h}_{j,k}$, dans lesquels on substitue chaque occurrence de xy par $u-1$, et qu'on évalue en $(u\mathrm{U}, x\mathrm{X}, y\mathrm{Y})$. Voyons cela sur un exemple.

Exemple pour $m = 2, t = 1$. La famille de monômes est $\mathrm{M} = \{1, x, u, x^2, xu, u^2, yu^2\}$ et $\ell = 7$. La famille de polynômes $\bar{g}_{i,k}$ donne $e^2, xe^2, x^2e^2, \bar{f}e, x\bar{f}e, \bar{f}^2$. On a un seul polynôme $\bar{h}_{j,k}$: $y\bar{f}^2$. On a $y\bar{f}^2 = yu^2 + 2\mathrm{A}uxy + \mathrm{A}^2x^2y$. On substitue chaque occurrence de xy par $u-1$ dans ce polynôme ce qui donne $yu^2 + 2\mathrm{A}u(u-1) + \mathrm{A}^2x(u-1) = yu^2 + 2\mathrm{A}u^2 - 2\mathrm{A}u + \mathrm{A}^2xu - \mathrm{A}^2x$. Après substitution on obtient donc que ce polynôme est combinaison de monômes de M. On considère ensuite la matrice 7×7 suivante qui donnera la base du réseau $\mathscr{L}$:

$$\begin{array}{c} \\ e^2 \\ xe^2 \\ \bar{f}e \\ x^2e^2 \\ x\bar{f}e \\ \bar{f}^2 \\ y\bar{f}^2 \end{array}
\begin{array}{c} \begin{array}{ccccccc} 1 & x & u & x^2 & ux & u^2 & u^2y \end{array} \\
\begin{pmatrix}
e^2 & & & & & & \\
 & e^2\mathrm{X} & & & & & \\
 & e\mathrm{AX} & e\mathrm{U} & & & & \\
 & & & e^2\mathrm{X}^2 & & & \\
 & & & e\mathrm{AX}^2 & e\mathrm{UX} & & \\
 & & & \mathrm{A}^2\mathrm{X}^2 & 2\mathrm{AUX} & \mathrm{U}^2 & \\
 & -\mathrm{A}^2\mathrm{X} & -2\mathrm{AU} & & \mathrm{A}^2\mathrm{UX} & 2\mathrm{AU}^2 & \mathrm{U}^2\mathrm{Y}
\end{pmatrix} \end{array}$$

$\boxed{20}$ Montrer que dans le cas général les polynômes $\bar{g}_{i,k}$ et les polynômes $\bar{h}_{j,k}$, dans lesquels on substitue chaque occurrence de xy par $u-1$, sont combinaisons de monômes de M.

$\boxed{21}$ Écrire une fonction qui prend un entrée un polynôme de $\mathbf{Z}[x,y,u]$ et qui ressort le polynôme correspondant après substitution de chaque occurrence de xy par $u-1$. En déduire une fonction qui crée la matrice de base du réseau $\mathscr{L}$ étant donné les paramètres m et t, et N, e, δ tels qu'en question 16. On posera $X = \lceil e^{\delta} \rceil$, $Y = \lceil e^{1/2} \rceil$ et $U = 1 + XY$.

$\boxed{22}$ Écrire une fonction qui prend un entrée un polynôme de $\mathbf{Z}[x,y,u]$ et qui ressort le polynôme correspondant de $\mathbf{Z}[x,y]$ après substitution de chaque occurrence de u par $1+xy$.

$\boxed{23}$ Soit $P \in \mathbf{Z}[x,y]$ un polynôme qui est à la somme d'au plus ω monômes. Supposons que $P(x_0,y_0) \equiv 0 \pmod{e^m}$ pour un entier m avec $|x_0| < X$ et $|y_0| < Y$ et que la norme du vecteur des coefficients du polynôme évalué en (xX, yY) satisfait $\|P(xX,yY)\| < e^m/\sqrt{\omega}$. Montrer que $P(x_0,y_0) = 0$ dans $\mathbf{Z}$.

Pour la suite de la cryptanalyse, on applique LLL à la base du réseau $\mathscr{L}$ et à partir de deux vecteurs, par exemple les deux premiers, on reconstruit deux polynômes $P_1, P_2 \in \mathbf{Z}[x,y]$ (en enlevant les bornes X, Y, U puis en substituant u par $1+xy$). Si ces deux polynômes satisfont les hypothèses de la question précédente, ils auront la même racine (x_0,y_0) dans $\mathbf{Z}$. Pour la retrouver, on calcule le résultant[3] de P_1 et P_2 par rapport à la variable y, et s'il est non nul, on cherche ses racines, ce qui doit donner la racine x_0. En réinjectant cette racine, dans P_1 on peut ensuite retrouver y_0.

$\boxed{24}$ En déduire une fonction qui prend en entrée N, e, δ tels qu'en question 16 et le paramètre m et qui tente de retrouver d et la factorisation de N. Pour t, on posera $t := \lfloor m(1-2\delta) \rfloor$. Donner des exemples d'applications pour N produit de deux nombres premiers de 1024 bits et $\delta = 0.2$ ($m = 2$ et $t = 1$ doivent être suffisants mais pas forcément avec les deux premiers vecteurs) et $\delta = 0.25$.

$\boxed{25}$ Récupérer dans le fichier

`https://www.math.u-bordeaux.fr/~gcastagn/Cryptanalyse/RSA.sage`

une clef publique (N, e) RSA ayant un petit exposant privé d. Retrouver d et la factorisation de N.

[3] Avec Sage : `P1.resultant(P2,y)`