Amélioration des attaques différentielles sur *KLEIN*

Cryptanalyse de la version complète de *KLEIN*-64

Virginie Lallemand
sous la direction de María Naya-Plasencia

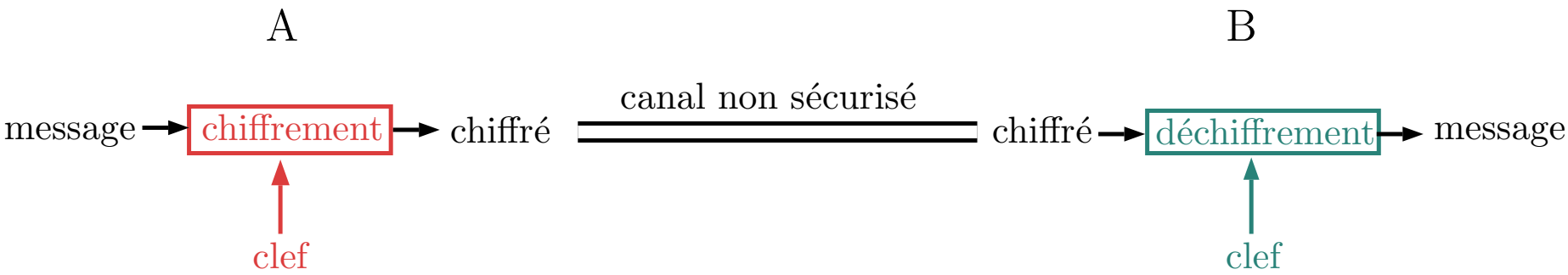
stage réalisé à Inria Paris-Rocquencourt
équipe SECRET

du 1er mars au 31 août 2013

Sommaire

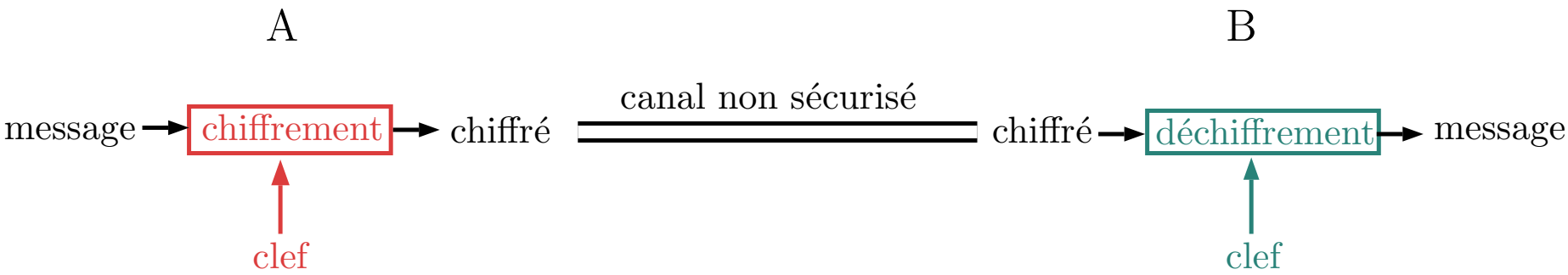
La Cryptographie

Un des buts principaux de la cryptographie est d'assurer la confidentialité de données transmises sur un canal non sûr



La Cryptographie

Un des buts principaux de la cryptographie est d'assurer la confidentialité de données transmises sur un canal non sûr



2 types d'algorithmes :

-symétriques

Partage d'une même clef par l'expéditeur et le destinataire

Même clef utilisée pour chiffrer et déchiffrer

-asymétriques

Chaque personne possède 2 clefs :

La clef *publique* du destinataire sert à l'expéditeur pour chiffrer son message

La clef *privée* du destinataire lui sert à déchiffrer

Nouveaux Enjeux

Nouveaux besoins cryptographiques nés de la nécessité de sécuriser les nouveaux supports miniaturisés (RFID, réseaux de capteurs ...) avec des algorithmes cryptographiques adaptés répondant à des critères stricts de :

- Place
- Rapidité
- Consommation d'énergie
- Sécurité

$\rightarrow$ **Cryptographie à bas coût** (*lightweight cryptography*)

Cryptographie à bas coût

- Algorithmes symétriques pour leur rapidité et leur efficacité d'implémentation matérielle et logicielle
- Les algorithmes existants ne répondent pas aux critères
- Exigences de sécurité moindres

Cryptographie à bas coût

- Algorithmes symétriques pour leur rapidité et leur efficacité d'implémentation matérielle et logicielle
- Les algorithmes existants ne répondent pas aux critères
- Exigences de sécurité moindres

$\rightarrow$ Nécessité de créer de nouvelles primitives

Propositions récentes : PRESENT, KATAN, LBlock, KLEIN ...

Problèmatique

- Contrairement à la cryptographie asymétrique, il n'existe pas de réduction de sécurité pour la cryptographie symétrique
- La confiance en un système est basée sur les meilleures analyses existantes

Problèmatique

- Contrairement à la cryptographie asymétrique, il n'existe pas de réduction de sécurité pour la cryptographie symétrique
- La confiance en un système est basée sur les meilleures analyses existantes

$\rightarrow$ Il est nécessaire de mettre à l'épreuve les cryptosystèmes pour s'assurer de leur fiabilité

Quelques Bases de Cryptographie Symétrique

Chiffrement par Bloc

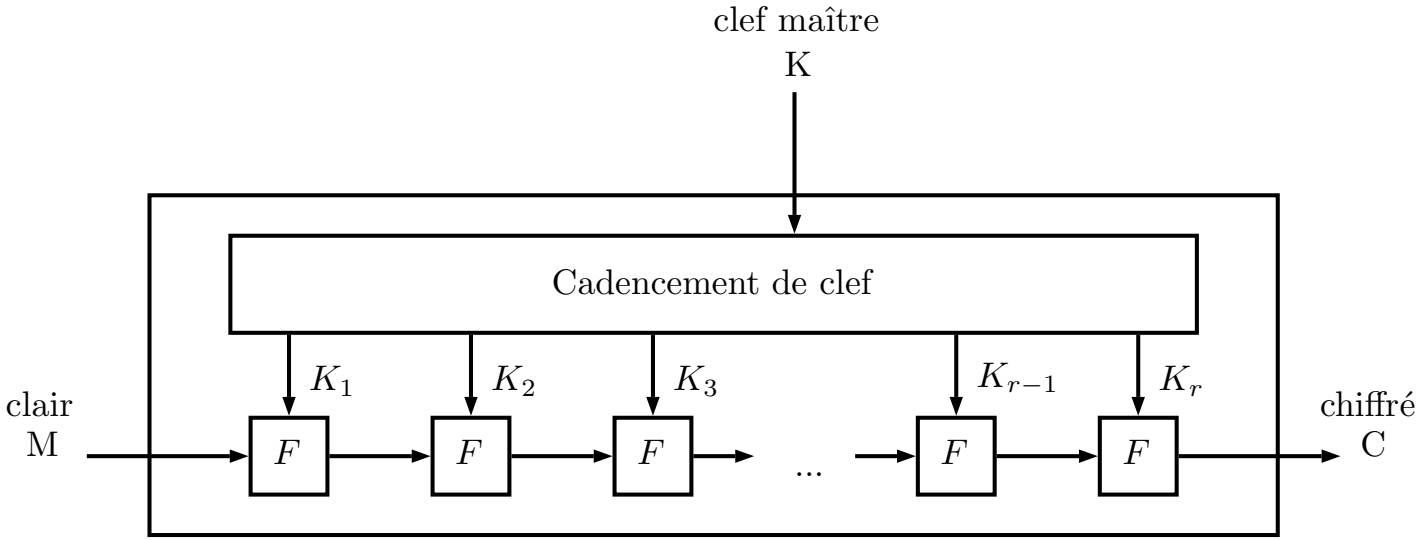
Chiffrement par bloc

Un chiffrement par bloc est une fonction E prenant en entrée 2 arguments : un bloc M de m bits et une clef secrète K de k bits et ayant comme sortie un bloc de m bits C.

$$\begin{array}{rcl} E : \{0,1\}^k \times \{0,1\}^m & \to & \{0,1\}^m \\ (K \quad , \quad M \quad) & \mapsto & c := E(K,M) = E_K(M) \end{array}$$

Chiffrement itératif

- r répétitions d'une même fonction F (*fonction de tour*)
- F paramétrée par une information dérivée de la clef : les *clefs de tour* $K_1 \cdots K_r$
- Celles-ci sont dérivées de la clef maître par un algorithme de cadencement de clef



Sécurité et Attaques

Objectif de sécurité visé : rendre toute recherche de la clef secrète au moins aussi lente que la recherche exhaustive. Dans le cas contraire, le système sera considéré comme *cassé*.

Sécurité et Attaques

Objectif de sécurité visé : rendre toute recherche de la clef secrète au moins aussi lente que la recherche exhaustive. Dans le cas contraire, le système sera considéré comme *cassé*.

Différents modèles d'attaques pour évaluer un système :

- À clair connu : l'attaquant possède des couples (clair,chiffré) aléatoires
- À clair choisi : il choisit les messages clairs et obtient les chiffrés correspondants
- À chiffré choisi : il peut obtenir le chiffrement et le déchiffrement des messages qu'il désire

Sécurité et Attaques

Objectif de sécurité visé : rendre toute recherche de la clef secrète au moins aussi lente que la recherche exhaustive. Dans le cas contraire, le système sera considéré comme *cassé*.

Différents modèles d'attaques pour évaluer un système :

- À clair connu : l'attaquant possède des couples (clair,chiffré) aléatoires
- À clair choisi : il choisit les messages clairs et obtient les chiffrés correspondants
- À chiffré choisi : il peut obtenir le chiffrement et le déchiffrement des messages qu'il désire

3 quantités définissent la performance d'une attaque :

- La complexité en données : nombre de messages nécessaires pour mener l'attaque
- La complexité en temps : nombre d'opérations nécessaires, calculé en nombre de chiffrements
- La complexité en mémoire : taille de stockage nécessaire

Principales Familles d'Attaques

- Algébriques
- Statistiques

Principe

Consiste à mettre en avant des comportements du système qui s'éloignent de ceux attendus d'une permutation aléatoire pour retrouver de l'information sur la clef.

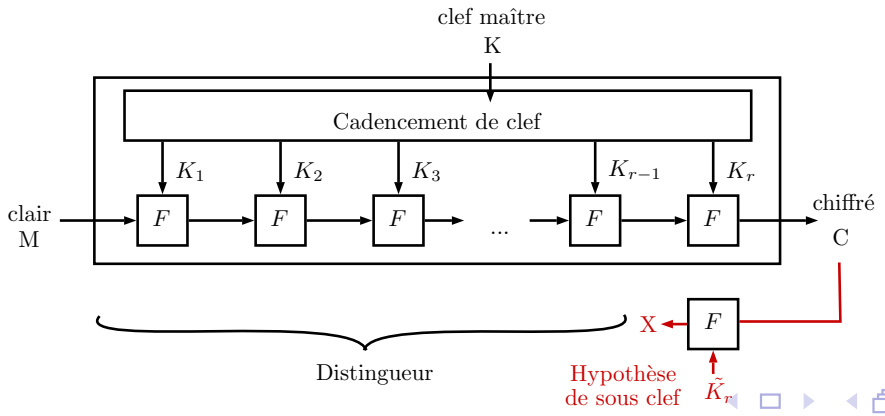
Distingueur

Définition

Un distingueur est un algorithme qui par un jeu de questions/réponses à une permutation arrive à décider avec probabilité supérieure à 1/2 s'il s'agit d'une permutation aléatoire ou s'il s'agit d'un chiffrement particulier.

Avec un distingueur des $r-1$ premiers tours, on peut monter une **attaque statistique sur le dernier tour** :

- Obtention des chiffrés
- Déchiffrement du dernier tour avec chacune des hypothèses de sous clef candidate.
- Utilisation du distingueur qui réagit si la sous clef testée est la bonne, alors qu'une mauvaise sous clef donne un résultat proche de celui d'une permutation aléatoire.



La Cryptanalyse Différentielle - Attaque de Base

Attaque statistique utilisant un biais dans la distribution des différences de paires de messages sur $r-1$ tours de la fonction de chiffrement

Définition

Une différentielle sur t tours d'un système de chiffrement est un couple $(\delta_{in}, \delta_{out}) \in \mathbb{F}_2^m \times \mathbb{F}_2^m$ constitué d'une différence en entrée et d'une différence en sortie après t tours.

Il existe le plus souvent plusieurs façons d'obtenir δ_{out} à partir de δ_{in} : chaque manière spécifique constitue un *chemin différentiel* :

Définition

Un chemin différentiel sur t tours d'un système de chiffrement itératif est un $(t+1)-$uplet $(\delta_{in} = \delta_0, \delta_2, \cdots, \delta_t = \delta_{out}) \in (\mathbb{F}_2^m)^{(t+1)}$ de différences intermédiaires à chaque sortie de tour.

$\rightarrow$ Un chemin différentiel sur $r-1$ tours possédant un biais significatif permet de construire un distingueur donc de mener une attaque sur le dernier tour

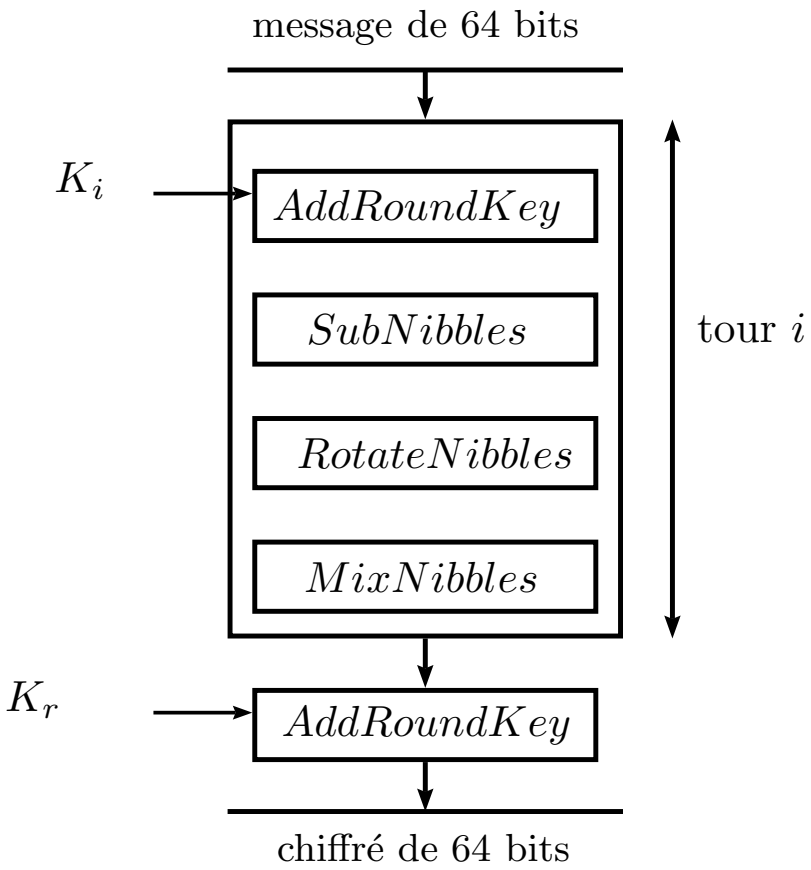
Le Chiffrement à Bas Coût KLEIN

Le chiffrement à bloc KLEIN

Famille de chiffrements par bloc à bas coût présentée en 2011 à la conférence RFIDsec par Zheng Gong, Svetla Nikova, et Yee-Wei Law

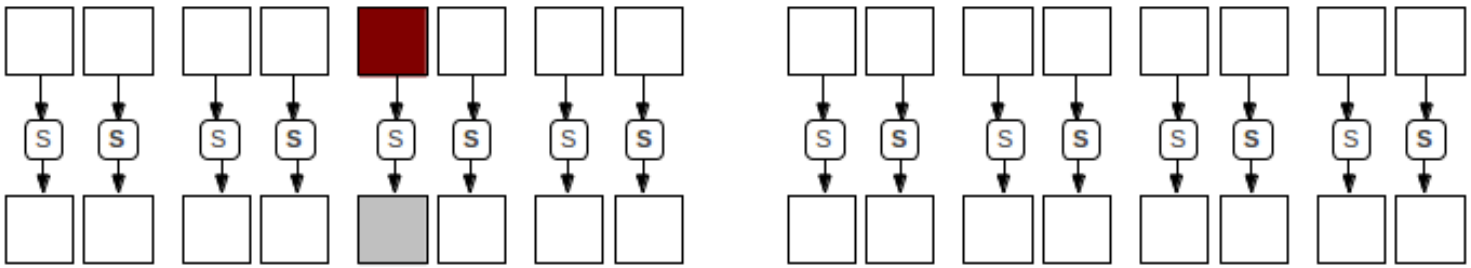
- Conçue pour une application software
- Chiffrement itératif de type Substitution-Permutation
- 3 versions : KLEIN-64, KLEIN-80 et KLEIN-96 réalisant 12, 16 et 20 tours

Description



Description

SubNibbles

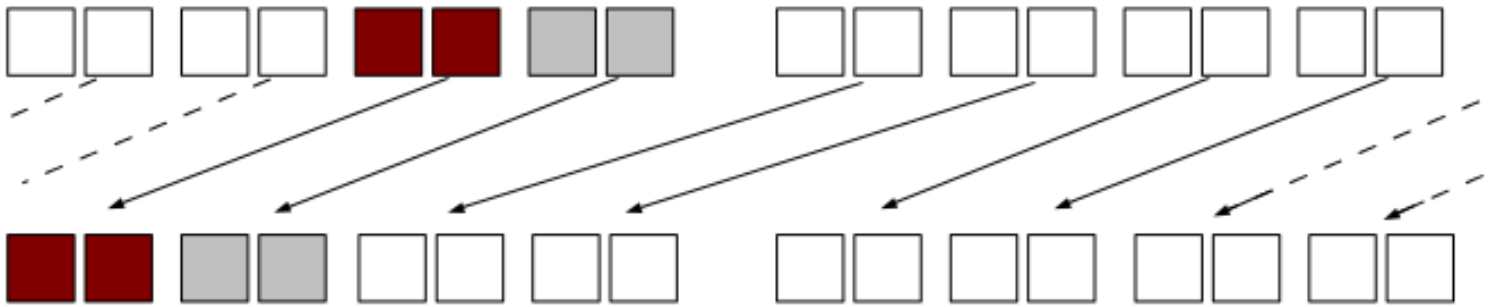


x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
S[x]	7	4	a	9	1	f	b	0	c	3	2	6	8	e	d	5

Description

RotateNibbles

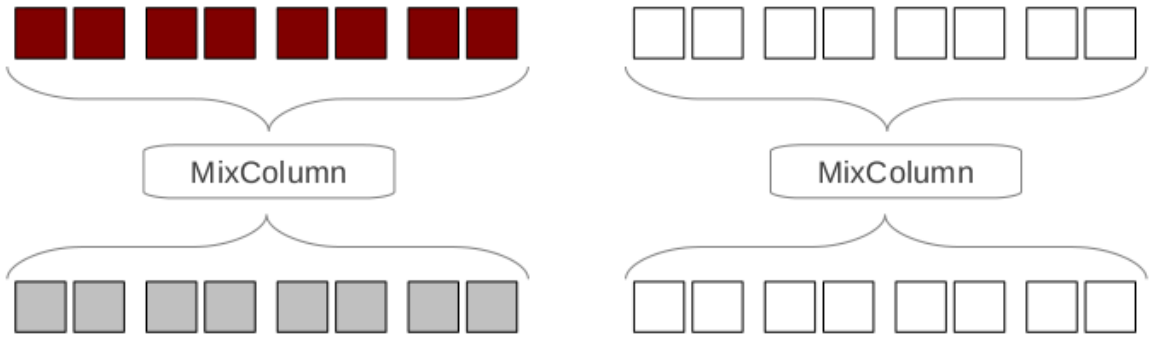
Effectue une rotation de l'état de 16 bits vers la gauche



Description

MixNibbles

Traite les 2 demi-états séparément avec l'opération *MixColumn* d'*AES*



MixColumn est réalisé en considérant chaque octet comme un élément de $GF(2^8) = GF(2)/(x^8 + x^4 + x^3 + x + 1)$ et en multipliant le vecteur colonne composé de 4 octets par la matrice suivante :

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}$$

Cryptanalyses précédentes

Version	Source	tours	Donnée	Temps	Mémoire	Type d'attaque
KLEIN-64	[Yu, Wu, Li, Zhang, *Inscrypt11*]	7	$2^{34.3}$	$2^{45.5}$	2^{32}	intégrale
	[Yu, Wu, Li, Zhang, *Inscrypt11*]	8	2^{32}	$2^{46.8}$	2^{16}	tronquée
	[Aumasson, Naya-Plasencia, Saarinen, *Indocrypt11*]	8	2^{35}	2^{35}	-	tronquée
	[Ahmadian, Salmasizadeh, Reza Aref *ePrint iacr 2013*]	12	2^{39}	$2^{62.84}$	$2^{4.5}$	biclique
KLEIN-80	[Yu, Wu, Li, Zhang, *Inscrypt11*]	8	$2^{34.3}$	$2^{77.5}$	2^{32}	intégrale

- Aucune attaque n'a encore été proposée sur KLEIN-96
- L'analyse biclique ne peut pas être considérée comme une attaque puisque qu'elle se résume à une recherche exhaustive accélérée de la clef

Idées principales [Yu et al, Inscrypt11], [Aumasson et al, Indocrypt11]

proposition

Toutes les opérations de tour sauf *MixNibbles* agissent indépendamment sur les quartets. De plus, les quartets bas restent en position basse et les quartets hauts restent en position haute.

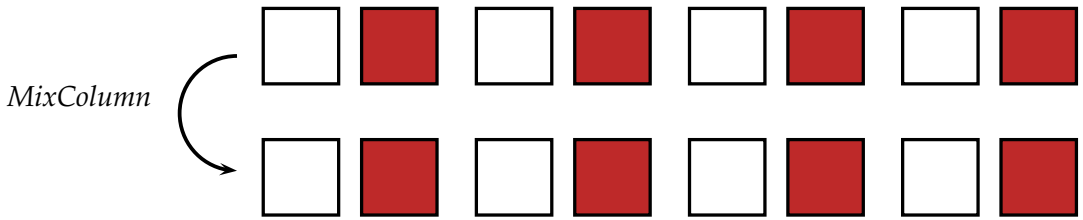
Idées principales [Yu et al, Inscrypt11], [Aumasson et al, Indocrypt11]

proposition

Toutes les opérations de tour sauf *MixNibbles* agissent indépendamment sur les quartets. De plus, les quartets bas restent en position basse et les quartets hauts restent en position haute.

proposition

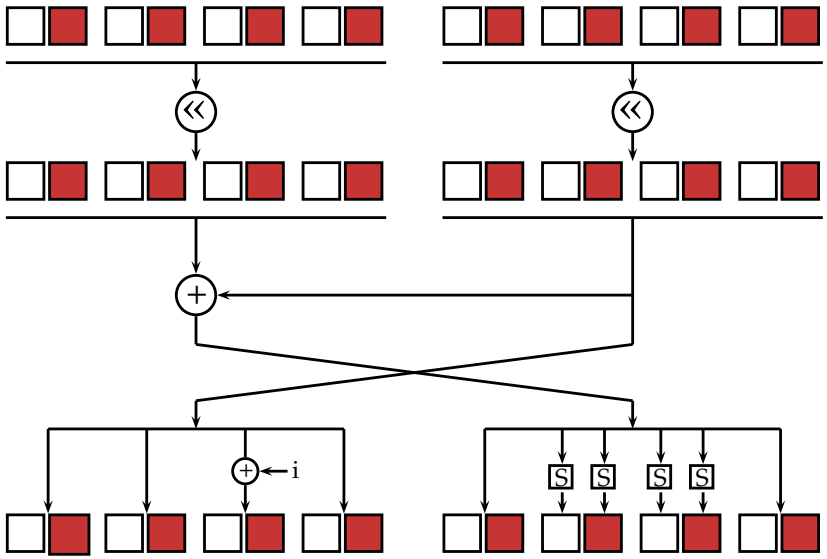
Si la différence en entrée de *MixColumn* est de la forme '0X0X0X0X' où X désigne un quartet de valeur quelconque et "0" désigne le quartet nul $(0000)_2$ alors la différence de sortie est de la même forme avec probabilité 2^{-3}.



Idées principales [Yu et al, Inscrypt11], [Aumasson et al, Indocrypt11]

proposition

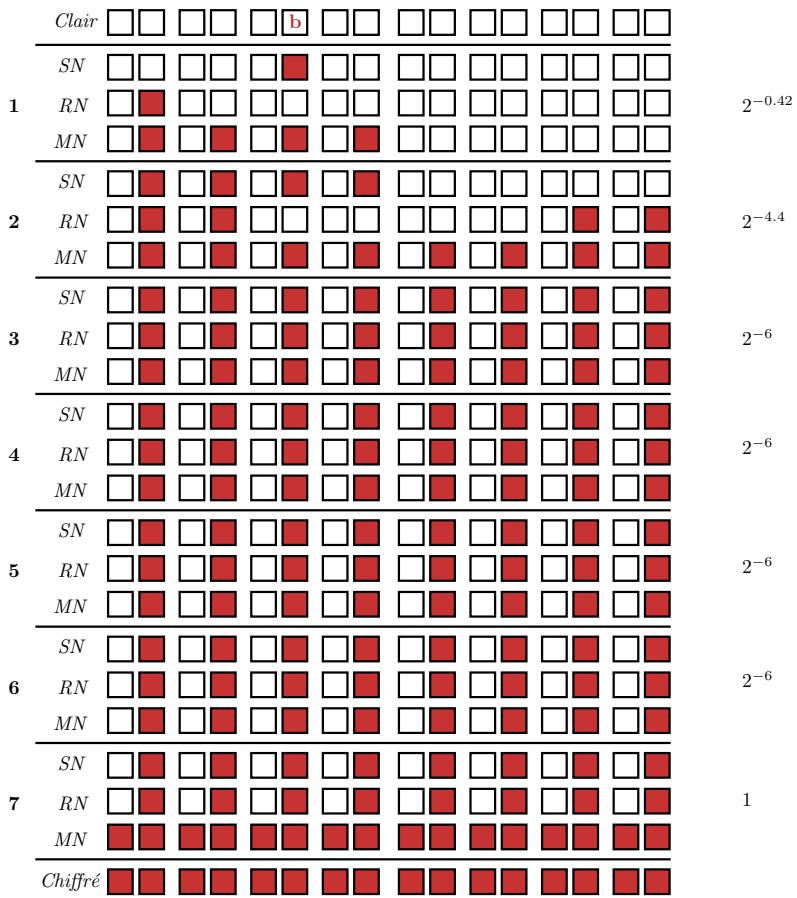
Lors du cadencement de clefs, les quartets bas et hauts ne sont pas mélangés : il est possible de calculer les quartets bas d'une clef de tour en possédant uniquement les quartets bas d'une des autres clefs de tour.



[Aumasson, Naya-Plasencia, Saarinen, *Indocrypt11*]

Attaque sur 7 tours

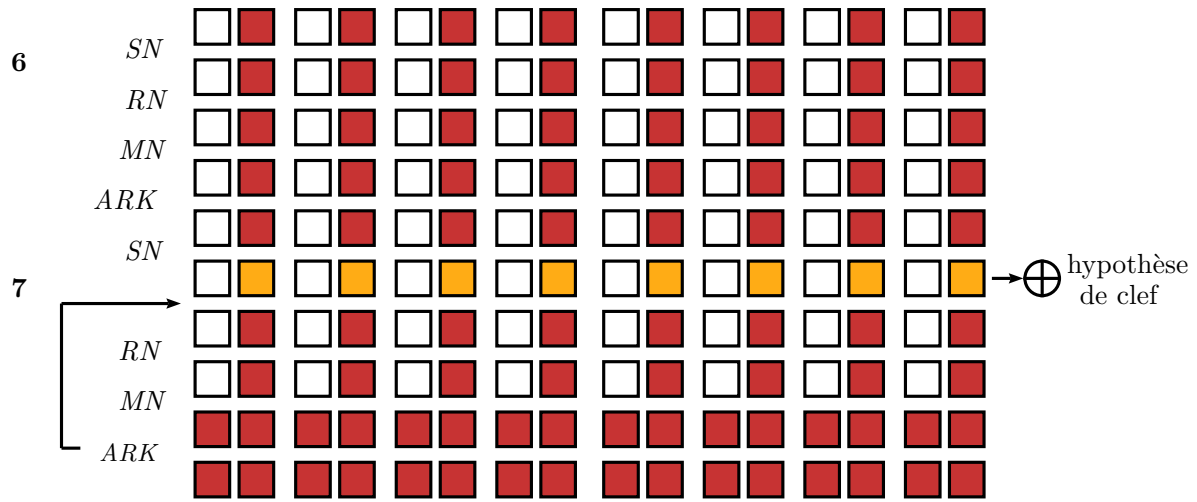
- Chemin différentiel de probabilité $2^{-28.08}$
- Clef maître retrouvée en 2^{33} opérations



[Aumasson, Naya-Plasencia, Saarinen, *Indocrypt11*]

Procédure

1. Demander le chiffrement de $2^{28.08}$ paires de messages
2. Conserver la paire qui vérifie le chemin différentiel
3. Effectuer une hypothèse sur la clef
4. Tester cette hypothèse en inversant en différence l'étape MixNibbles du tour 6 (probabilité 2^{-6})



Efficacité et Limitations

pour 7 tours :

- il est facile de trouver une bonne paire puisque celle-ci possède les quartets hauts inactifs lorsqu'on inverse le dernier MN à partir du chiffré. De plus, cette observation est de probabilité $2^{-28.08}$ ce qui est supérieur au cas générique d'obtenir 32 bits de différence à 0 (2^{-32}) $\Rightarrow$ Une paire avec des quartets hauts inactifs au dernier tour est une bonne paire
- Chaque bonne paire permet d'obtenir un filtre de 6 bits
- En répétant la procédure avec plusieurs bonnes paires on parvient à la bonne valeur des quartets bas

Efficacité et Limitations

pour 7 tours :

- il est facile de trouver une bonne paire puisque celle-ci possède les quartets hauts inactifs lorsqu'on inverse le dernier MN à partir du chiffré. De plus, cette observation est de probabilité $2^{-28.08}$ ce qui est supérieur au cas générique d'obtenir 32 bits de différence à 0 (2^{-32}) $\Rightarrow$ Une paire avec des quartets hauts inactifs au dernier tour est une bonne paire
- Chaque bonne paire permet d'obtenir un filtre de 6 bits
- En répétant la procédure avec plusieurs bonnes paires on parvient à la bonne valeur des quartets bas

Si on élargit l'attaque à 8 tours :

- Apparition de fausses alarmes
- Une technique dite des *bits neutres* est nécessaire

Nouvelle Attaque

Nouvelle Attaque

Idée clef

Problématique

Notre optique est d'augmenter le nombre de tours attaqués

Nouvelle Attaque

Idée clef

Problématique

Notre optique est d'augmenter le nombre de tours attaqués

Mais dans ce cas ci :

$\rightarrow$ Le chemin différentiel sera de très faible probabilité

$\rightarrow$ On aura besoin de beaucoup de données pour trouver 1 bonne paire

Nouvelle Attaque

Idée clef

Problématique

Notre optique est d'augmenter le nombre de tours attaqués

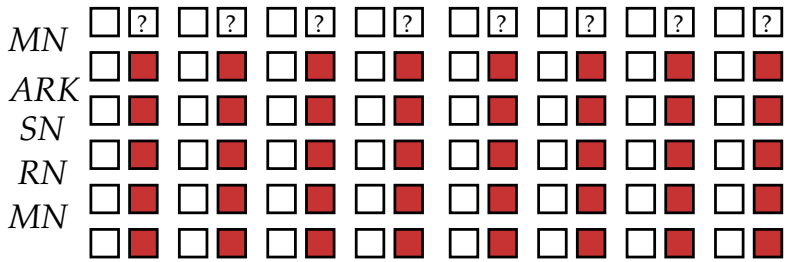
Mais dans ce cas ci :

$\rightarrow$ Le chemin différentiel sera de très faible probabilité

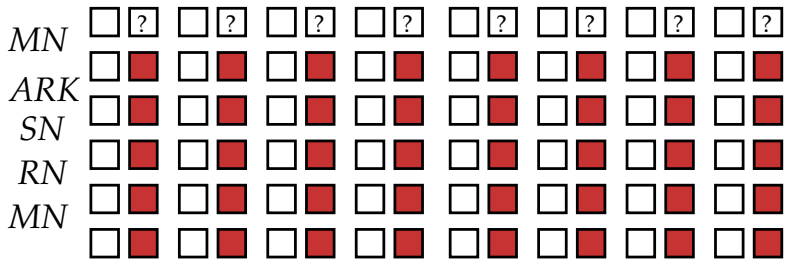
$\rightarrow$ On aura besoin de beaucoup de données pour trouver 1 bonne paire

- L'attaque précédente utilise l'avant dernière opération MixNibble pour décider si une clef est possible ou non et répète la procédure pour filtrer suffisamment
- Notre idée consiste à chercher une bonne paire et à filtrer les clefs candidates avec toutes les étapes *MixNibbles* précédentes

Comment inverser un tour?

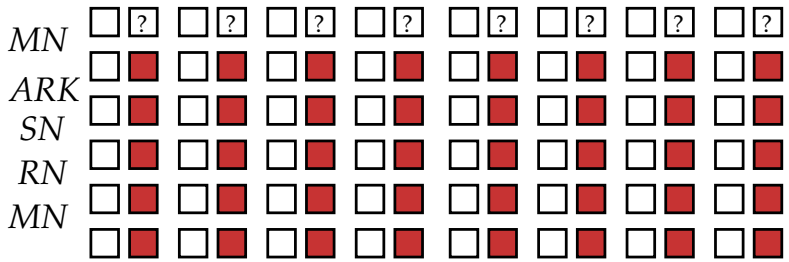


Comment inverser un tour?



$\Rightarrow$ On souhaite obtenir la différence en entrée de l'opération MixNibbles précédente

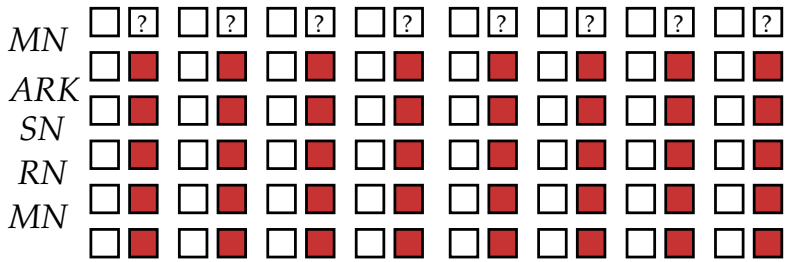
Comment inverser un tour?



$\Rightarrow$ On souhaite obtenir la différence en entrée de l'opération MixNibbles précédente

- Inverser un tour entier donc MN, RN, SN et ARK

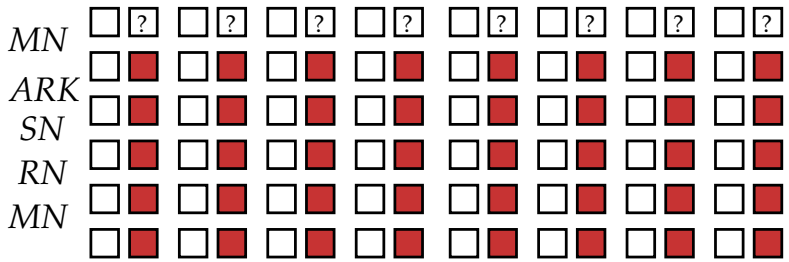
Comment inverser un tour?



$\Rightarrow$ On souhaite obtenir la différence en entrée de l'opération MixNibbles précédente

- Inverser un tour entier donc MN, RN, SN et ARK
- Seulement besoin des quartets bas pour inverser RN, SN and ARK

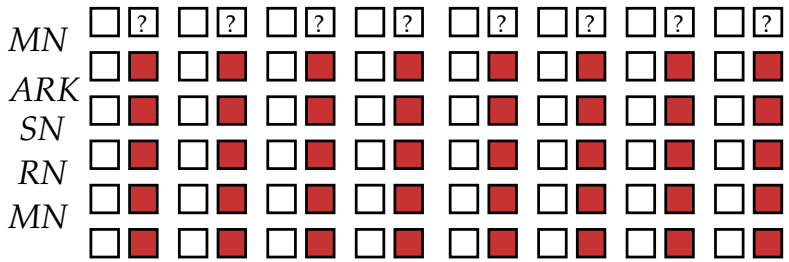
Comment inverser un tour?



$\Rightarrow$ On souhaite obtenir la différence en entrée de l'opération MixNibbles précédente

- Inverser un tour entier donc MN, RN, SN et ARK
- Seulement besoin des quartets bas pour inverser RN, SN and ARK
- Nécessité de connaître les quartets bas de la clef pour inverser ARK

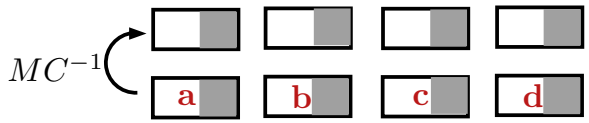
Comment inverser un tour?



$\Rightarrow$ On souhaite obtenir la différence en entrée de l'opération MixNibbles précédente

- Inverser un tour entier donc MN, RN, SN et ARK
- Seulement besoin des quartets bas pour inverser RN, SN and ARK
- Nécessité de connaître les quartets bas de la clef pour inverser ARK
- Comment inverser MC qui agit sur 32 bits à la fois?

Inversion de MN



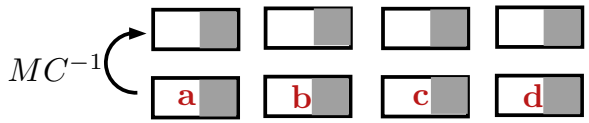
soit $a = (a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)$ l'écriture binaire de l'octet a.

proposition

Pour obtenir les quartets bas résultant de l'inversion de l'opération *MixColumn* connaissant les quartets bas de l'entrée (a, b, c, d), il est nécessaire de connaître 3 valeurs dépendant des quartets hauts inconnus :

$$\begin{cases} a_1 + a_2 + b_2 + c_0 + c_1 + c_2 + d_0 + d_2 \\ a_1 + b_0 + b_1 + c_1 + d_0 + d_1 \\ a_0 + a_1 + a_2 + b_0 + b_2 + c_1 + c_2 + d_2 \end{cases}$$

Inversion de MN



soit $a = (a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)$ l'écriture binaire de l'octet a.

proposition

Pour obtenir les quartets bas résultant de l'inversion de l'opération *MixColumn* connaissant les quartets bas de l'entrée (a, b, c, d), il est nécessaire de connaître 3 valeurs dépendant des quartets hauts inconnus :

$$\begin{cases} a_1 + a_2 + b_2 + c_0 + c_1 + c_2 + d_0 + d_2 \\ a_1 + b_0 + b_1 + c_1 + d_0 + d_1 \\ a_0 + a_1 + a_2 + b_0 + b_2 + c_1 + c_2 + d_2 \end{cases}$$

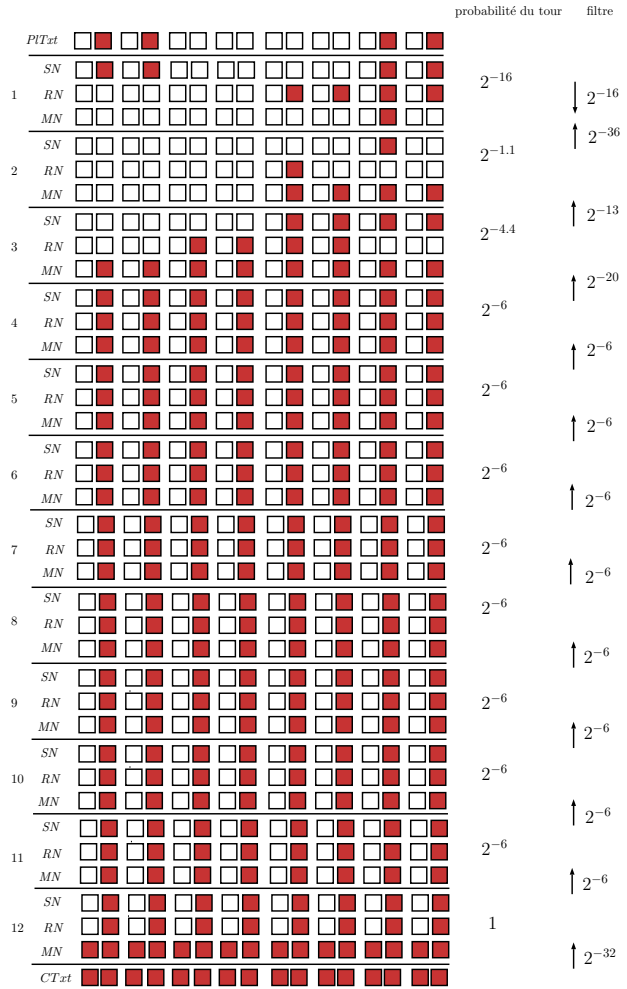
$\Rightarrow$ On peut prédire les quartets bas avec une hypothèse de 6 bits

Notre Nouvelle Attaque sur KLEIN-64

Considérer le chemin différentiel ci-contre de probabilité $2^{-69.5}$

1. Obtenir les chiffrés correspondant aux paires de différence d'entrée voulue
2. Conserver les paires ayant les quartets bas inactifs au dernier tour

   Pour chacune de ces paires :
3. Effectuer une hypothèse sur les quartets bas de la clef et utiliser les conditions du premier tour pour les filtrer
4. Inverser un tour en quartets bas avec une hypothèse de 6 bits
5. Vérifier que la différence obtenue en entrée du précédent MN est comme voulue : 1 hypothèse parmi les 2^6 passe
6. Si les conditions sont vérifiées, répéter la procédure avec un autre tour



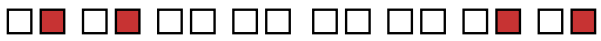
Réussite de l'attaque

- Lorsque tous les tours ont été inversés, il reste 2^8 candidats
- Une recherche exhaustive des quartets hauts de la clef permet d'écarter les mauvais candidats
- Nous avons proposé d'autres compromis donnée/mémoire/temps grâces à d'autres chemins différentiels

Optimisations

Optimisation 1

Utiliser des structures pour limiter le nombre de données nécessaires



Structure

Une structure est l'ensemble des messages que l'on peut former en faisant varier toutes les positions de différence autorisées.

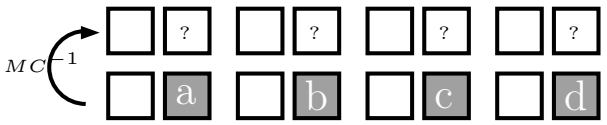
Ici une structure correspond à fixer 48 bits et à faire varier les 4 quartets bas restants, soit 2^{16} chiffrements pour 1 structure qui permet de construire $\frac{2^{16} \times (2^{16}-1)}{2} \simeq 2^{31}$ paires.
Pour obtenir les $2^{69.5}$ paires nécessaires à l'attaque, on aura seulement besoin de $2^{54.5}$ chiffrements.

Optimisations

proposition

On obtient un état de sortie dont les quartets bas sont nuls étant donnée une entrée du même type ssi les égalités suivantes sont vérifiées :

$$\begin{cases} a_4 + b_4 + c_4 + d_4 = 0 \\ a_4 + a_5 + b_5 + c_4 + c_5 + d_5 = 0 \\ a_4 + a_5 + a_6 + b_4 + b_6 + c_5 + c_6 + d_6 = 0 \end{cases}$$

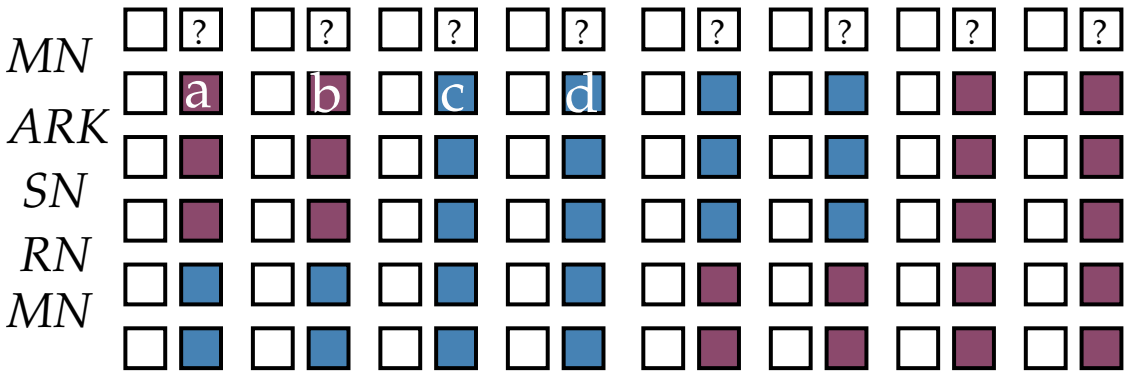


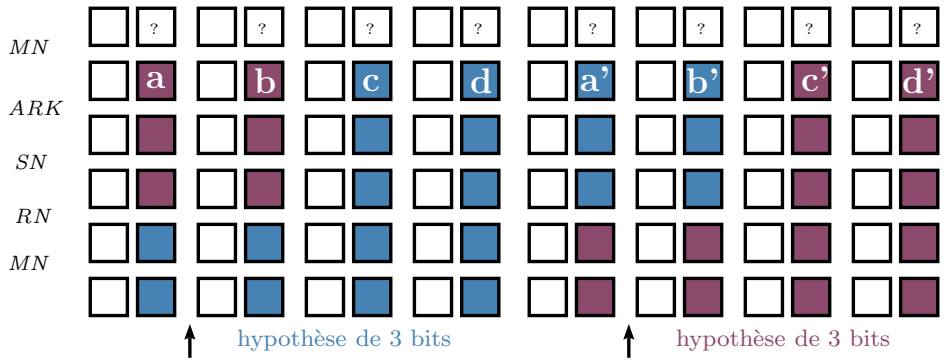
Optimisations

proposition

On obtient un état de sortie dont les quartets bas sont nuls étant donnée une entrée du même type ssi les égalités suivantes sont vérifiées :

$$\begin{cases} a_4 + b_4 + c_4 + d_4 = 0 \\ a_4 + a_5 + b_5 + c_4 + c_5 + d_5 = 0 \\ a_4 + a_5 + a_6 + b_4 + b_6 + c_5 + c_6 + d_6 = 0 \end{cases}$$

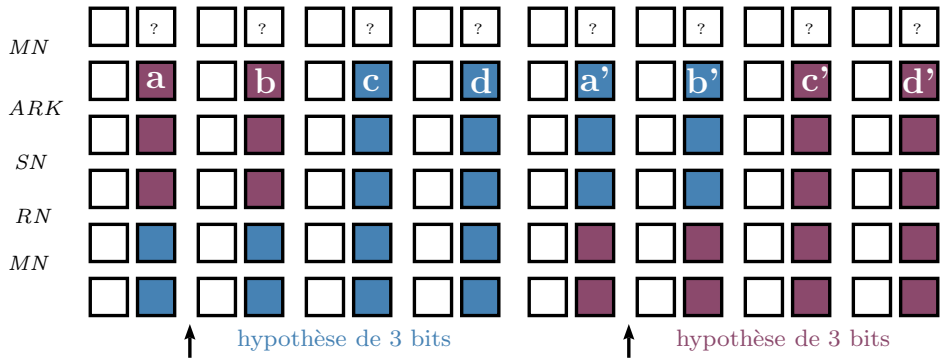




On crée une liste contenant les 6 bits suivants pour chacune des 2^3 hypothèses possibles :

$$a_4 + b_4 \qquad c'_4 + d'_4$$
$$a_4 + a_5 + b_5 \qquad c'_4 + c'_5 + d'_5 \qquad 2^3 \text{ calculs}$$
$$a_4 + a_5 + a_6 + b_4 + b_6 \qquad c'_5 + c'_6 + d'_6$$

$$c_4 + d_4 \qquad a'_4 + b'_4$$
$$c_4 + c_5 + d_5 \qquad a'_4 + a'_5 + b'_5 \qquad 2^3 \text{ calculs}$$
$$c_5 + c_6 + d_6 \qquad a'_4 + a'_5 + a'_6 + b'_4 + b'_6$$



On crée une liste contenant les 6 bits suivants pour chacune des 2^3 hypothèses possibles :

$$a_4 + b_4 \qquad c'_4 + d'_4$$
$$a_4 + a_5 + b_5 \qquad c'_4 + c'_5 + d'_5 \qquad 2^3 \text{ calculs}$$
$$a_4 + a_5 + a_6 + b_4 + b_6 \qquad c'_5 + c'_6 + d'_6$$

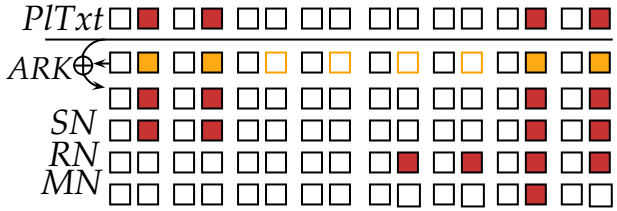
$$c_4 + d_4 \qquad a'_4 + b'_4$$
$$c_4 + c_5 + d_5 \qquad a'_4 + a'_5 + b'_5 \qquad 2^3 \text{ calculs}$$
$$c_5 + c_6 + d_6 \qquad a'_4 + a'_5 + a'_6 + b'_4 + b'_6$$

On essaye de combiner les 2 pour obtenir les égalités précédentes (probabilité 2^{-6})
$\rightarrow 2^4$ chiffrements de tour au total

Optimisations

Optimisation 3

Utiliser l'indépendance des 2 demi-états dans l'opération MixNibbles pour réduire le nombre d'opérations à effectuer lors des hypothèses de clef du premier tour



Effectuer une hypothèse de 16 bits de clef à la fois
$\rightarrow$ On obtient les quartets bas de clef possibles avec 2^{16} chiffrements de tour contre 2^{32} avec la version naïve

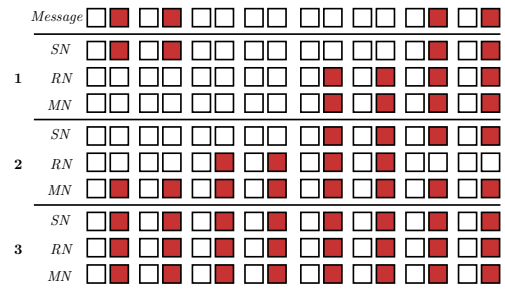
Optimisations

Optimisation 4

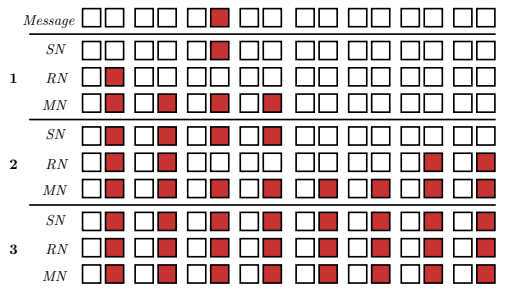
La recherche des quartets hauts peut être accélérée en utilisant l'information provenant des hypothèses de 6 bits faites lors des inversions

Complexité des différents cas sur la version complète de KLEIN-64

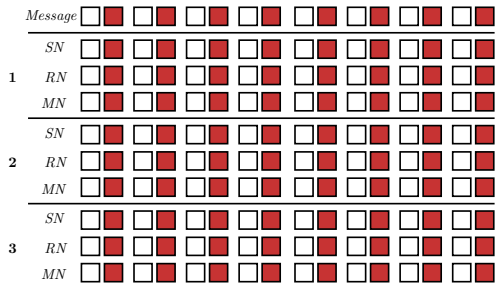
Cas	Données	Temps	Mémoire
1	$2^{54.5}$	2^{57}	2^{16}
2	$2^{56.5}$	2^{62}	2^{4}
3	2^{35}	$2^{63.8}$	2^{32}
4	2^{46}	2^{62}	2^{16}



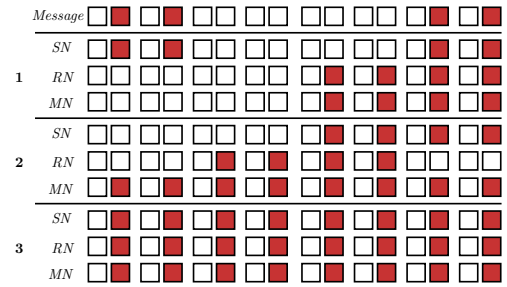
(a) cas 1



(b) cas 2



(c) cas 3



(d) cas 4

Complexité de l'attaque sur les autres versions de KLEIN

KLEIN-80 et KLEIN-96 possèdent plus d'itération de la fonction de tour et possèdent des clefs plus longues :

→ Nécessite plus d'hypothèses puisque clefs plus longues

→ Plus de tour donc plus de difficultés à atteindre la version complète

Version	Cas	Tours	Données	Temps	Mémoire
80	1	13	$2^{60.49}$	$2^{71.1}$	2^{16}
80	2	13	$2^{62.49}$	2^{76}	2^{4}
80	3	13	2^{41}	2^{78}	2^{32}
80	4	13	2^{52}	$2^{71.3}$	2^{16}
96	3	14	2^{47}	2^{92}	2^{32}
96	4	14	2^{58}	$2^{89.2}$	2^{16}

Table: Complexités des différents cas pour KLEIN-80 et KLEIN-96

Conclusion

- Première attaque sur la version complète de KLEIN-64
- Validée expérimentalement sur une version réduite à 9 tours
- Approuvée par les concepteurs de KLEIN
- Permet d'atteindre 13 tours sur les 16 de KLEIN-80 et 14 tours sur les 20 de KLEIN-96
- Plusieurs changements pourraient rendre KLEIN plus résistant : changer la matrice de *MixNibbles*, effectuer un RotateNibbles d'un nombre de bits non multiple de 8, changer le KeySchedule ...