# *Sur les fonctions de hachage cryptographiques basées sur des graphes*

*On graph-based cryptographic hash functions*

Christophe Petit

# Cryptographie

- Pour les espions, mais aussi
  - email
  - gsm
  - e-virements
  - e-health
  - ...

# *Alice, Bob et Charlie*

# *Alice, Bob et Charlie*

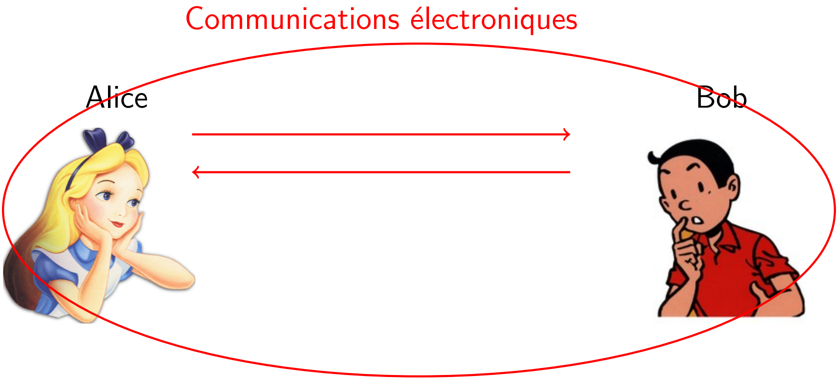
Alice

# *Alice, Bob et Charlie*

Alice

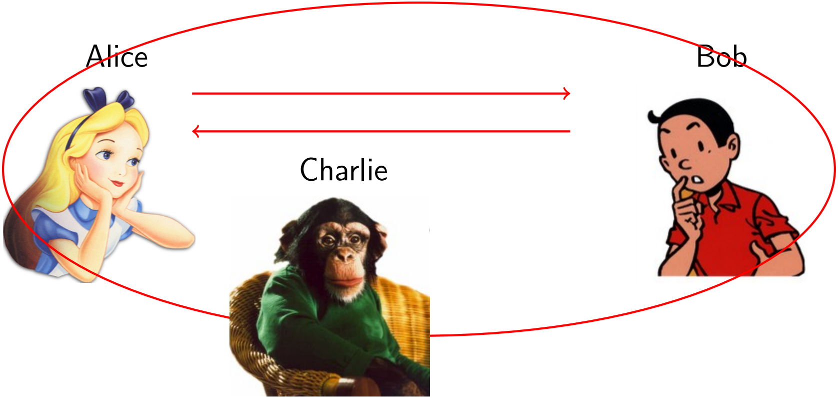
Bob

# *Alice, Bob et Charlie*



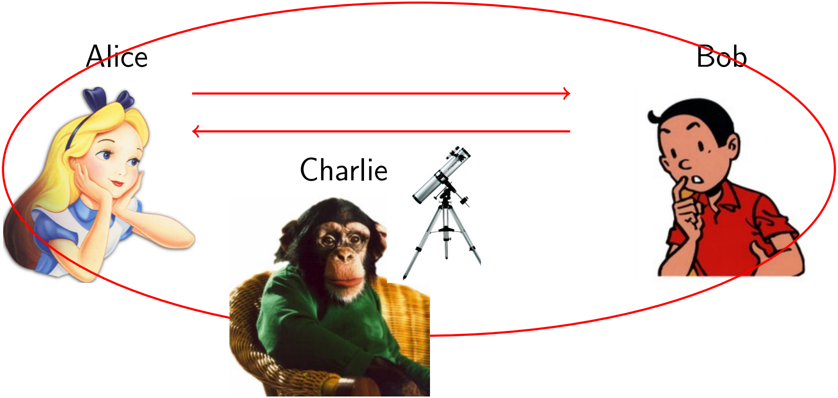
# *Alice, Bob et Charlie*

Communications électroniques

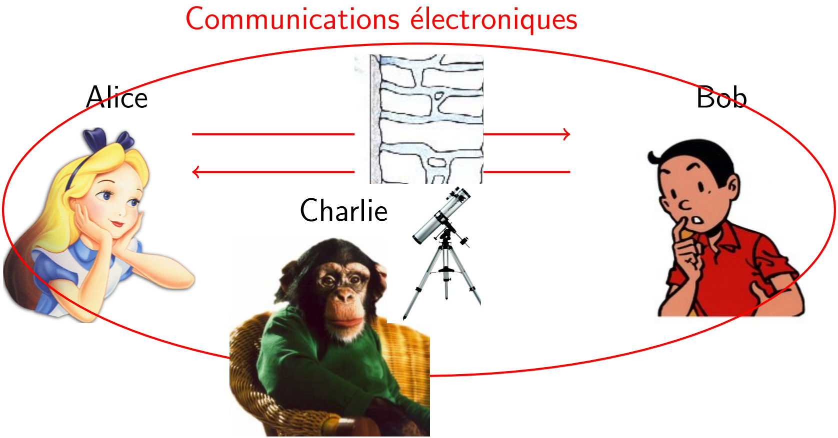


# *Alice, Bob et Charlie*

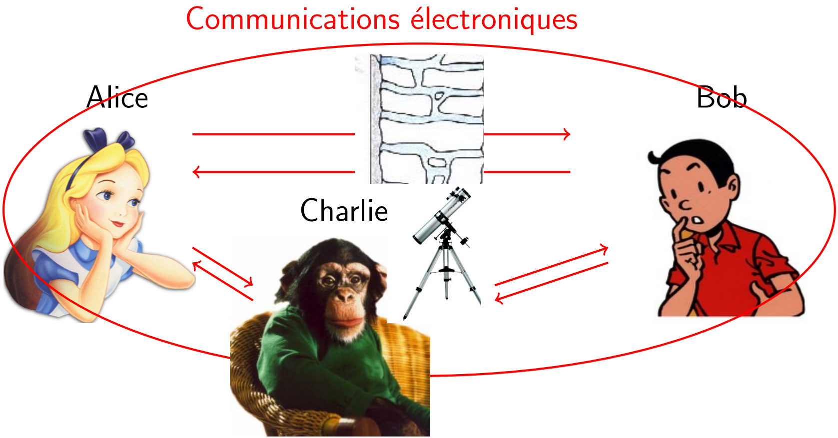
Communications électroniques



# *Alice, Bob et Charlie*



# *Alice, Bob et Charlie*



# *Fonctions de hachage*

# *Fonctions de hachage*

- Utilisées partout en crypto pour garantir
  - Intégrité
  - Authenticité
  - Confidentialité

# Fonctions de hachage

- Utilisées partout en crypto pour garantir
  - Intégrité
  - Authenticité
  - Confidentialité

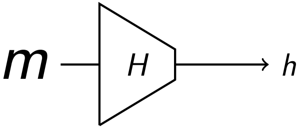
  via MACs, signatures, dérivation de clés, stockage de mots de passe, certains schémas de chiffrement,...

# *Fonctions de hachage*

- Utilisées partout en crypto pour garantir
  - Intégrité
  - Authenticité
  - Confidentialité

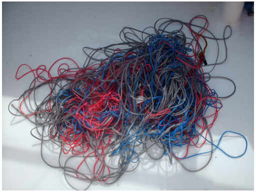
  via MACs, signatures, dérivation de clés, stockage de mots de passe, certains schémas de chiffrement,...

- **Compressent** leurs entrées



# *Fonctions de hachage basées sur des graphes*

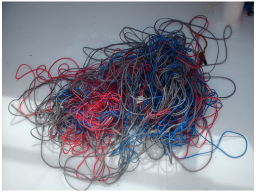
- Importante structure mathématique basée sur (certains) graphes

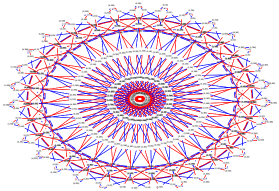
  Fonction de hachage classique

# *Fonctions de hachage basées sur des graphes*

- Importante structure mathématique basée sur (certains) graphes

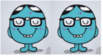


Fonction de hachage classique

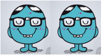


Fonction de hachage basée sur un graphe

# *Plan de l'exposé*

- Introduction 
- Motivations 
- Construction et attaques génériques 
- Quelques résultats de la thèse 
- Conclusion 

# *Plan de l'exposé*

- Introduction 
- Motivations 
- Construction et attaques génériques 
- Quelques résultats de la thèse 
- Conclusion 

# Problème d'authentification

# *Problème d'authentification*

Essaye ce super jeu !

$\longrightarrow$

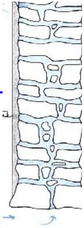

# Problème d'authentification

Essaye ce super jeu !

$\longrightarrow$

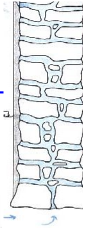

# *Problème d'authentification*

Essaye ce super

# *Problème d'authentification*

Essaye ce super

Essaye ce super jeu !


# *Problème(s) d'authentification*

- Besoin d'authentifier
  - Le message
  - L'expéditeur

# *Problème(s) d'authentification*

- Besoin d'authentifier
  - Le message
  - L'expéditeur
- Deux solutions classiques:
  - **Signatures**
  - **MACs**

# *Problème(s) d'authentification*

- Besoin d'authentifier
  - Le message
  - L'expéditeur
- Deux solutions classiques:
  - **Signatures** à partir d'un secret possédé par Alice
  - **MACs**

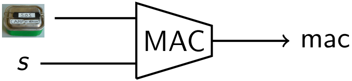
# *Problème(s) d'authentification*

- Besoin d'authentifier
  - Le message
  - L'expéditeur

- Deux solutions classiques:
  - **Signatures** à partir d'un secret possédé par Alice
  - **MACs** à partir d'un secret partagé par Alice et Bob

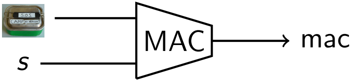
# *Solution 1: MACs*

- Alice et Bob ont une **clé secrète** $s$ et utilisent un MAC:
  - Alice calcule



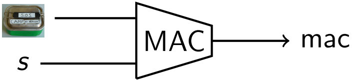
# *Solution 1: MACs*

- Alice et Bob ont une **clé secrète** *s* et utilisent un MAC:
  - Alice calcule



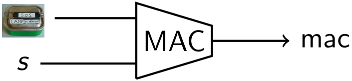

  - Alice envoie le résultat avec le message

# *Solution 1: MACs*

- Alice et Bob ont une **clé secrète** *s* et utilisent un MAC:
  - Alice calcule 

  - Alice envoie le résultat avec le message
  - Quand il reçoit un message, Bob recalcule le résultat et compare

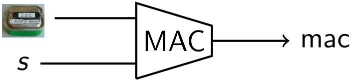
# *Solution 1: MACs*

- Alice et Bob ont une **clé secrète** *s* et utilisent un MAC:
  - Alice calcule 

  - Alice envoie le résultat avec le message
  - Quand il reçoit un message, Bob recalcule le résultat et compare
- Intuition: même si Charlie peut modifier le message, il ne peut pas calculer un MAC valide car il ne connaît pas *s*

# *Solution 1: MACs*

- Alice et Bob ont une **clé secrète** *s* et utilisent un MAC:
  - Alice calcule




  - Alice envoie le résultat avec le message
  - Quand il reçoit un message, Bob recalcule le résultat et compare
- Intuition: même si Charlie peut modifier le message, il ne peut pas calculer un MAC valide car il ne connaît pas *s*
- Souvent construits à partir de **fonctions de hachage**

# *Solution 2: Signatures digitales*

- Alice a une **clé privée** à laquelle correspond une autre clé, publique
  - Alice signe le message avec sa clé privée
  - Elle envoie la signature avec le message
  - Tout le monde peut vérifier la signature avec la clé publique

# *Solution 2: Signatures digitales*

- Alice a une **clé privée** à laquelle correspond une autre clé, publique
  - Alice signe le message avec sa clé privée
  - Elle envoie la signature avec le message
  - Tout le monde peut vérifier la signature avec la clé publique
- Intuition: impossible de produire de fausses signatures

# *Signatures digitales*

+ Il existe de bons algorithmes de signature (ex. RSA)

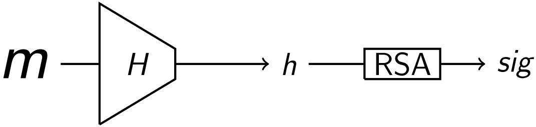
# *Signatures digitales*

+ Il existe de bons algorithmes de signature (ex. RSA)

– Tels quels, ces algorithmes sont
  - trop lents pour de longs messages
  - vulnérables à des attaques simples exploitant leur structure mathématique

# *Signatures digitales*

+ Il existe de bons algorithmes de signature (ex. RSA)
– Tels quels, ces algorithmes sont
    - trop lents pour de longs messages
    - vulnérables à des attaques simples exploitant leur structure mathématique
- Solution: "hash-then-sign paradigm"



# *Applications des fonctions de hachage*

- MACs
- Signatures digitales

# *Applications des fonctions de hachage*

- MACs
- Signatures digitales
- Stockage de mots de passe

# Applications des fonctions de hachage

- MACs
- Signatures digitales
- Stockage de mots de passe
- Generation de nombres pseudo-aléatoires

# Applications des fonctions de hachage

- MACs
- Signatures digitales
- Stockage de mots de passe
- Generation de nombres pseudo-aléatoires
- Extraction d'entropie

# Applications des fonctions de hachage

- MACs
- Signatures digitales
- Stockage de mots de passe
- Generation de nombres pseudo-aléatoires
- Extraction d'entropie
- Techniques de dérivations de clés

# *Applications des fonctions de hachage*

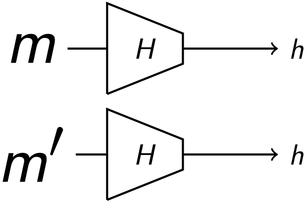
- MACs
- Signatures digitales
- Stockage de mots de passe
- Generation de nombres pseudo-aléatoires
- Extraction d'entropie
- Techniques de dérivations de clés
- ...
- ...

# *Applications des fonctions de hachage*

- MACs
- Signatures digitales
- Stockage de mots de passe
- Generation de nombres pseudo-aléatoires
- Extraction d'entropie
- Techniques de dérivations de clés
- ...
- ...

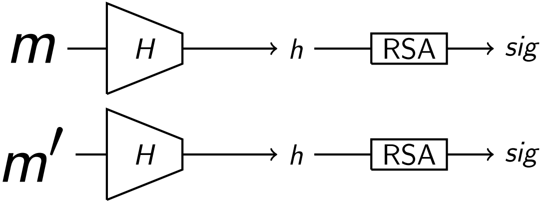
# Propriétés des fonctions de hachage

- Propriétés principales:
  - **Résistance aux collisions:** "dur" de calculer $m, m'$ tels que $H(m) = H(m')$



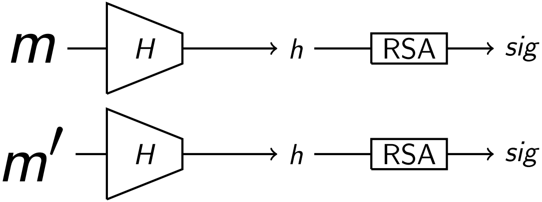
# *Propriétés des fonctions de hachage*

- Propriétés principales:
  - **Résistance aux collisions:** “dur” de calculer $m, m'$ tels que $H(m) = H(m')$



# *Propriétés des fonctions de hachage*

- Propriétés principales:
  - **Résistance aux collisions:** "dur" de calculer $m, m'$ tels que $H(m) = H(m')$




  - **Résistance aux préimages:** étant donné la valeur $H(m)$ pour un certain $m$, "dur" de trouver $m'$ tel que $H(m) = H(m')$

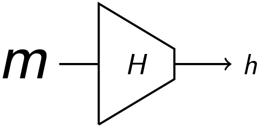
# *Propriétés des fonctions de hachage*

- Propriétés principales:
  - **Résistance à la seconde préimage:** étant donné $m$, "dur" de trouver $m' \neq m$ tel que $H(m) = H(m')$

# *Propriétés des fonctions de hachage*

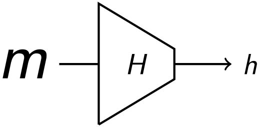
- Propriétés principales:
  - **Résistance à la seconde préimage:** étant donné $m$, "dur" de trouver $m' \neq m$ tel que $H(m) = H(m')$
  - **Compression:** $H : \{0,1\}^* \rightarrow \{0,1\}^\lambda$

    ($\lambda \approx 160$ en pratique)



# *Propriétés des fonctions de hachage*

- Propriétés principales:
  - **Résistance à la seconde préimage:** étant donné $m$, "dur" de trouver $m' \neq m$ tel que $H(m) = H(m')$
  - **Compression:** $H : \{0,1\}^* \rightarrow \{0,1\}^\lambda$




($\lambda \approx 160$ en pratique)

  - Distribution presque **uniforme** en sortie

# *Propriétés des fonctions de hachage*

- Autres propriétés:
  - XOR résistance
  - ADD résistance
  - Non-multiplicativité
  - “Oracle aléatoire”

# *Comment "prouver" qu'on a ces propriétés?*

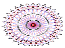
- En crypto, on prouve la **sécurité par réductions:**

  **si** une certaine propriété du système n'est pas vérifiée,
  **alors** on peut résoudre un certain problème "difficile"

# *Comment "prouver" qu'on a ces propriétés?*

- En crypto, on prouve la **sécurité par réductions:**

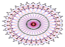
  **si** une certaine propriété du système n'est pas vérifiée,
  **alors** on peut résoudre un certain problème "difficile"

- La *sécurité* de l'algorithme est *déduite* de la difficulté de casser certaines briques de base
  -  problèmes mathématiques (IFP, DLP, ECDLP,...)
  -  autres algorithmes crypto (AES, SHA,...)

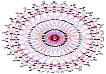
# *Comment "prouver" qu'on a ces propriétés?*

- En crypto, on prouve la **sécurité par réductions:**

  **si** une certaine propriété du système n'est pas vérifiée,
  **alors** on peut résoudre un certain problème "difficile"

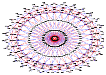
- La *sécurité* de l'algorithme est *déduite* de la difficulté de casser certaines briques de base
  -  problèmes mathématiques (IFP, DLP, ECDLP,...)
  -  autres algorithmes crypto (AES, SHA,...)

- Design et évaluation réduits

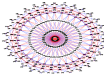
 *vs.* 

- Problèmes mathématiques: structure claire

- AES, SHA: structure complexe

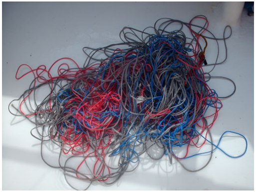
 *vs.* 

- Problèmes mathématiques: structure claire
  - − Plus facile à casser a priori
  - + Meilleure confiance si résistance
  - + (Certains) bien étudiés, même hors crypto
  - + Faciles à maintenir
  - − Ne peut pas tout prouver, autres faiblesses
- AES, SHA: structure complexe

 *vs.* 

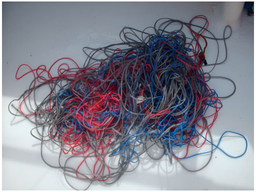
- Problèmes mathématiques: structure claire
  - − Plus facile à casser a priori
  - + Meilleure confiance si résistance
  - + (Certains) bien étudiés, même hors crypto
  - + Faciles à maintenir
  - − Ne peut pas tout prouver, autres faiblesses
- AES, SHA: structure complexe
  - + "Fonctions aléatoires", pas de faiblesse apparente
    $\rightarrow$ utiles comme couteaux suisses
  - ± Certains sont très bien étudiés (dans leur contexte initial)
  - − Durs à maintenir
  - − Structure mal comprise: faiblesses inconnues ?
  - + Souvent plus rapides

# *Fonctions de hachage basées sur des graphes*

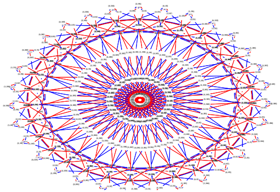


SHA & fonctions de hachage classiques: sécurité basée sur des arguments heuristiques

# *Fonctions de hachage basées sur des graphes*



SHA & fonctions de hachage classiques: sécurité basée sur des arguments heuristiques



Fonctions de hachage basées sur des graphes: résistance aux collisions dépend de problèmes mathématiques

# *But et résultats de la thèse: étude des fonctions de hachage basées sur des graphes*

- **Sécurité**
  - Sécurité du design en général Ch.4
  - Sécurité des constructions particulières Ch.5,6,7,D
  - Malléabilité Ch.8

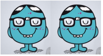
# *But et résultats de la thèse: étude des fonctions de hachage basées sur des graphes*

- **Sécurité**
  - Sécurité du design en général Ch.4
  - Sécurité des constructions particulières Ch.5,6,7,D
  - Malléabilité Ch.8
- **Efficacité**
  - Efficacité hardware et software des différentes constructions Ch.4,7,9
  - Algorithmes améliorés Ch.4,9,C
  - Parallélisme Ch.4,8,9

# *But et résultats de la thèse: étude des fonctions de hachage basées sur des graphes*

- **Sécurité**
  - Sécurité du design en général Ch.4
  - Sécurité des constructions particulières Ch.5,6,7,D
  - Malléabilité Ch.8
- **Efficacité**
  - Efficacité hardware et software des différentes constructions Ch.4,7,9
  - Algorithmes améliorés Ch.4,9,C
  - Parallélisme Ch.4,8,9
- **Applications**
  - Modélisation et utilisation de la malleabilité Ch.8
  - Suppression de la malléabilité Ch.9

# *Plan de l'exposé*

- Introduction 
- Motivations 
- Construction et attaques génériques 
- Quelques résultats de la thèse 
- Conclusion 

# *Construction*

- Intuition: à partir d'un graphe $k$-régulier (dirigé),

# Construction

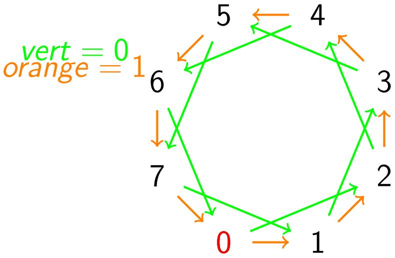
- Intuition: à partir d'un graphe $k$-régulier (dirigé), le message $m$ est écrit en base $k$: $m = m_1 m_2 ... m_\mu$

# *Construction*

- Intuition: à partir d'un graphe $k$-régulier (dirigé), le message $m$ est écrit en base $k$: $m = m_1 m_2 ... m_\mu$ et les $m_i$ fixent un chemin dans le graphe

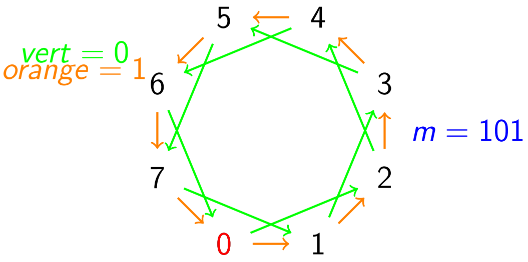
# Construction

- Intuition: à partir d'un graphe $k$-régulier (dirigé), le message $m$ est écrit en base $k$: $m = m_1 m_2 \ldots m_\mu$ et les $m_i$ fixent un chemin dans le graphe
- Colorier les arêtes avec $k$ couleurs associées à $0, \ldots, k-1$, choisir un sommet initial



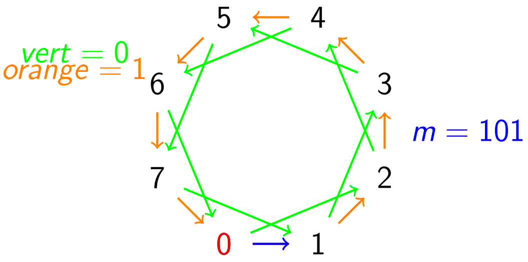
# Construction

- Intuition: à partir d'un graphe $k$-régulier (dirigé), le message $m$ est écrit en base $k$: $m = m_1 m_2 \ldots m_\mu$ et les $m_i$ fixent un chemin dans le graphe
- Colorier les arêtes avec $k$ couleurs associées à $0, \ldots, k-1$, choisir un sommet initial



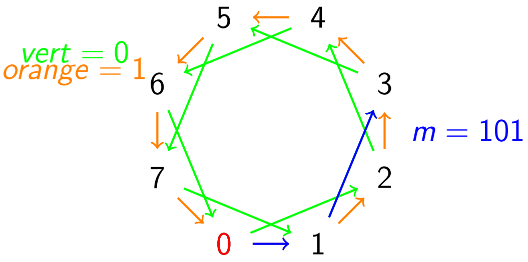
# *Construction*

- Intuition: à partir d'un graphe $k$-régulier (dirigé), le message $m$ est écrit en base $k$: $m = m_1 m_2 \ldots m_\mu$ et les $m_i$ fixent un chemin dans le graphe
- Colorier les arêtes avec $k$ couleurs associées à $0, \ldots, k-1$, choisir un sommet initial



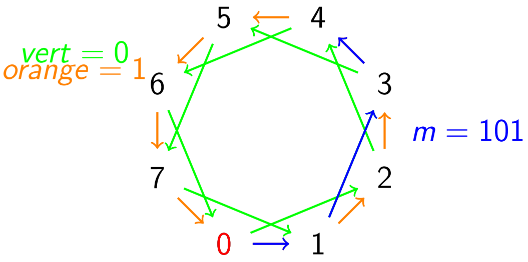
# Construction

- Intuition: à partir d'un graphe $k$-régulier (dirigé), le message $m$ est écrit en base $k$: $m = m_1 m_2 \ldots m_\mu$ et les $m_i$ fixent un chemin dans le graphe
- Colorier les arêtes avec $k$ couleurs associées à $0, \ldots, k-1$, choisir un sommet initial



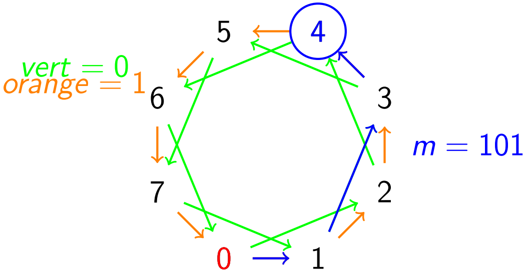
# Construction

- Intuition: à partir d'un graphe $k$-régulier (dirigé), le message $m$ est écrit en base $k$: $m = m_1 m_2 \ldots m_\mu$ et les $m_i$ fixent un chemin dans le graphe
- Colorier les arêtes avec $k$ couleurs associées à $0, \ldots, k-1$, choisir un sommet initial



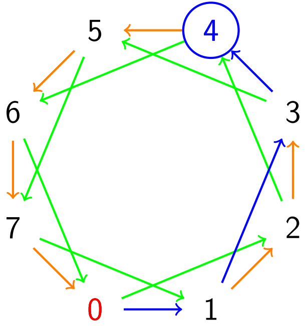
# *Construction*

- Intuition: à partir d'un graphe $k$-régulier (dirigé), le message $m$ est écrit en base $k$: $m = m_1 m_2 \ldots m_\mu$ et les $m_i$ fixent un chemin dans le graphe
- Colorier les arêtes avec $k$ couleurs associées à $0, \ldots, k-1$, choisir un sommet initial



# *Collisions et préimages*

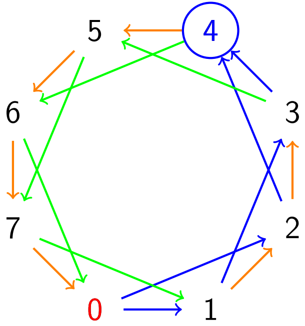
- Trouver une *préimage* revient à trouver *un chemin* de l'origine vers le sommet donné




( En crypto, $\geq 2^{160}$ sommets )

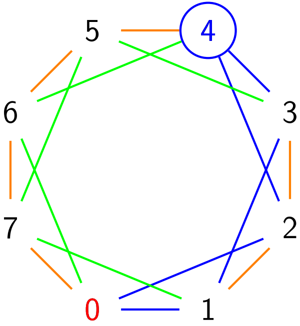
# *Collisions et préimages*

- Trouver une *collision* revient à trouver *deux chemins* partant à l'origine et finissant au même sommet



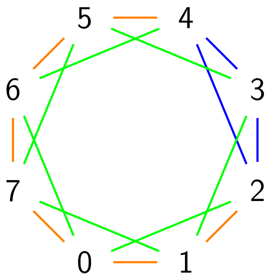
# *Collisions et préimages*

- Si le graphe est non dirigé, cela revient à trouver un *cycle* passant par l'origine



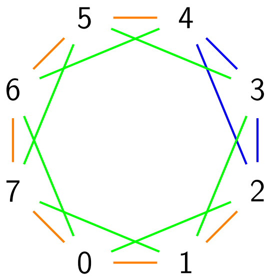
# *Maille*

- $\approx$ taille du plus petit cycle



# *Maille*

- ≈ taille du plus petit cycle




- Donne la plus petite “distance” entre toute paire de collisions

# *Expansion*

- Graphe d'expansion $\approx$ graphe avec très peu d'arêtes mais très bien connecté
  *"Réseau social efficace pour propager des ragots"*

# *Expansion*

- Graphe d'expansion $\approx$ graphe avec très peu d'arêtes mais très bien connecté
  *"Réseau social efficace pour propager des ragots"*
- Messages aléatoires de taille fixée:
  distribution des hachés $\longrightarrow$ distribution uniforme quand taille $\nearrow$

# *Expansion*

- Graphe d'expansion $\approx$ graphe avec très peu d'arêtes mais très bien connecté
  *"Réseau social efficace pour propager des ragots"*
- Messages aléatoires de taille fixée:
  distribution des hachés $\longrightarrow$ distribution uniforme quand taille $\nearrow$
- Convergence rapide
  ssi "grand" paramètre d'expansion

# *Fonctions de hachage basées sur des graphes: propriétés de sécurité*

| **fonction** | **graphe** |
|---|---|
| collisions | cycle/ double chemin |
| préimage | chemin |
| distribution des hachés | expansion |
| "distance" minimale de collision | maille |

# Graphe de Cayley

- Groupe: ensemble $G$ avec une loi interne
$$\cdot : G \times G \longrightarrow G$$
élement neutre, inverse, associativité

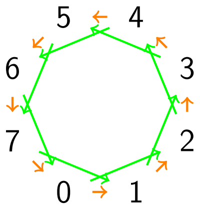
# Graphe de Cayley

- Groupe: ensemble $G$ avec une loi interne
$$\cdot : G \times G \longrightarrow G$$
élement neutre, inverse, associativité
- Graphe de Cayley $\mathcal{C}_{G,S} = (V, E)$:
pour un *groupe* $G$ et $S \subset G$,
  - un sommet $v_g$ pour chaque $g \in G$
  - une arête $(v_{g_1}, v_{g_2})$ ssi $\exists s \in S$ tel que $g_2 = g_1 \cdot s$

# Graphe de Cayley

- Groupe: ensemble $G$ avec une loi interne
$$\cdot : G \times G \longrightarrow G$$
élement neutre, inverse, associativité
- Graphe de Cayley $\mathcal{C}_{G,S} = (V, E)$:
pour un *groupe* $G$ et $S \subset G$,
  - un sommet $v_g$ pour chaque $g \in G$
  - une arête $(v_{g_1}, v_{g_2})$ ssi $\exists s \in S$ tel que $g_2 = g_1 \cdot s$

Exemple:
$G = (\mathbb{Z}/8\mathbb{Z}, +)$,
$S = \{1, 2\}$



# *Fonctions de hachage Cayley*

- Utilise des graphes de Cayley

  Exemple pour $S = \{s_0, s_1\}$ et sommet d'origine 1:

$$H(11001) = s_1 \cdot s_1 \cdot s_0 \cdot s_0 \cdot s_1$$

# *Fonctions de hachage Cayley*

- Utilise des graphes de Cayley

  Exemple pour $S = \{s_0, s_1\}$ et sommet d'origine 1:

$$H(11001) = s_1 \cdot s_1 \cdot s_0 \cdot s_0 \cdot s_1$$

- Simplifie la définition et l'étude
- Parallélisme: $H(m_1||m_2) = H(m_1) \cdot H(m_2)$

# *Fonctions de hachage Cayley: propriétés de sécurité*

| **fonction** | **graphe** | **groupe** |
|---|---|---|
| collisions | cycle/ double chemin | représentation/ bi-factorisation |
| préimage | chemin | factorisation dans le groupe |
| distribution des hachés | expansion | constante de Kazhdan |
| "distance" minimale de collision | maille | |

# *Problème de représentation*

- Etant donné un groupe $G$ et $S = \{s_1, \ldots s_k\} \subset G$, trouver un produit

$$\prod_{1 \leq i \leq N} s_{\theta(i)}^{e_i} = 1$$

satisfaisant quelques contraintes supplémentaires

# *Problème de représentation*

- Etant donné un groupe $G$ et $S = \{s_1, \ldots s_k\} \subset G$, trouver un produit

$$\prod_{1 \leq i \leq N} s_{\theta(i)}^{e_i} = 1$$

satisfaisant quelques contraintes supplémentaires

- Remplacer 1 par un autre élément du groupe
$\rightarrow$ problème de factorisation

# *Problème de représentation*

- Etant donné un groupe $G$ et $S = \{s_1, ...s_k\} \subset G$, trouver un produit

$$\prod_{1 \le i \le N} s_{\theta(i)}^{e_i} = 1$$

satisfaisant quelques contraintes supplémentaires

- Remplacer 1 par un autre élément du groupe
$\longrightarrow$ problème de factorisation
- Difficulté dépend très fort de $G$ et $S$ !

# *Attaques génériques*

- Recherche exhaustive en temps $2^{\lambda}$
- Attaque des anniversaires en temps $2^{\lambda/2}$

# *Attaques génériques*

- Recherche exhaustive en temps $2^{\lambda}$
- Attaque des anniversaires en temps $2^{\lambda/2}$
- Attaques "meet-in-the-middle"
  - Préimages en temps $2^{\lambda/2}$
  - Parce que chaque pas est inversible

# *Attaques génériques*

- Recherche exhaustive en temps $2^{\lambda}$
- Attaque des anniversaires en temps $2^{\lambda/2}$
- Attaques "meet-in-the-middle"
  - Préimages en temps $2^{\lambda/2}$
  - Parce que chaque pas est inversible
- Attaques de multicollisions
  - $t$-collisions en temps $\log_2 t 2^{\lambda/2}$ [Joux04]
  - A cause de la structure itérative

# *Attaques génériques*

- Recherche exhaustive en temps $2^{\lambda}$
- Attaque des anniversaires en temps $2^{\lambda/2}$
- Attaques "meet-in-the-middle"
  - Préimages en temps $2^{\lambda/2}$
  - Parce que chaque pas est inversible
- Attaques de multicollisions
  - $t$-collisions en temps $\log_2 t 2^{\lambda/2}$ [Joux04]
  - A cause de la structure itérative
- Attaques "trapdoor"
  - Via choix du sommet initial et/ou des paramètres du graphe

# *Attaques génériques*

- Attaques par sous-groupes sur les fonctions de hachage de Cayley

# *Attaques génériques*

- Attaques par sous-groupes sur les fonctions de hachage de Cayley
- Malléabilité
  - Fonctions de hachage de Cayley: pour tout $m, m'$

$$H(m||m') = H(m) \cdot H(m')$$

  - En général: étant donnés $H(m)$ et $m'$, facile de calculer $H(m||m')$...

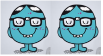
# *Attaques génériques*

- Attaques par sous-groupes sur les fonctions de hachage de Cayley
- Malléabilité
  - Fonctions de hachage de Cayley: pour tout $m, m'$

$$H(m||m') = H(m) \cdot H(m')$$

  - En général: étant donnés $H(m)$ et $m'$, facile de calculer $H(m||m')$... même si $m$ lui-même ne peut être calculé à partir de $H(m)$ !

# *Plan de l'exposé*

- Introduction 
- Motivations 
- Construction et attaques génériques 
- Quelques résultats de la thèse 
- Conclusion 

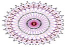
# Résultats principaux

- Fonctions de LPS et Morgenstern: [PLQ08]
  extension d'une attaque sur collision contre LPS à
  - Attaque sur préimage contre LPS
  - Attaques sur collisions et préimages contre Morgenstern

# *Résultats principaux*

- Fonctions de LPS et Morgenstern: [PLQ08]
  extension d'une attaque sur collision contre LPS à
  - Attaque sur préimage contre LPS
  - Attaques sur collisions et préimages contre Morgenstern
- Fonction de Zémor-Tillich: [PQTZ09]
  - Review des attaques existantes
  - Nouvelles attaques sur préimage et collision
  - Parties facile et difficile du problème de collisions
  - Introduction de deux variantes

# Résultats principaux

- Fonctions de LPS et Morgenstern: [PLQ08]
  extension d'une attaque sur collision contre LPS à
  - Attaque sur préimage contre LPS
  - Attaques sur collisions et préimages contre Morgenstern
- Fonction de Zémor-Tillich: [PQTZ09]
  - Review des attaques existantes
  - Nouvelles attaques sur préimage et collision
  - Parties facile et difficile du problème de collisions
  - Introduction de deux variantes
- ZesT: [PVQ08,PdMQTVZ09]
  - Nouvelle fonction de hachage basée sur Zémor-Tillich
  - Combine les avantages de  et de 

# *Fonction de hachage LPS*

- Construction: graphes LPS [LPS88, CGL07] (Cayley)
  - Soit $l$ premier et petit, $p$ premier et grand,
    $p \equiv l \equiv 1 \bmod 4$, $\left(\frac{l}{p}\right) = 1$
    Soit $\mathbf{i}$ tel que $\mathbf{i}^2 = -1 \bmod p$
  - Soit $G = PSL(2, \mathbb{F}_p)$,
    Soit $S = \{s_j, j = 1 ... l + 1\}$, où

$$s_j = \begin{pmatrix} \alpha_j + \mathbf{i}\beta_j & \gamma_j + \mathbf{i}\delta_j \\ -\gamma_j + \mathbf{i}\delta_j & \alpha_j - \mathbf{i}\beta_j \end{pmatrix}, \qquad j = 0, ..., l;$$

et $(\alpha_j, \beta_j, \gamma_j, \delta_j)$ sont toutes les solutions entières de $\alpha^2 + \beta^2 + \gamma^2 + \delta^2 = l$, avec $\alpha > 0$ et $\beta$, $\gamma$, $\delta$

# *Collisions pour la fonction LPS [TZ08]*

- Idée de Tillich et Zémor : **relever le problème de représentation** de $PSL(2,\mathbb{F}_p)$ vers $\Omega \subset SL(2,\mathbb{Z}[i])$:

$$\xleftarrow[\text{mod } p]{}$$

$$\begin{array}{rcl}
\mathbf{i}^2 = -1 & \rightarrow & i^2 = -1 \\
\mathbb{F}_p & \rightarrow & \mathbb{Z}[i] \\
PSL(2,\mathbb{F}_p) & \rightarrow & \color{blue}{\Omega \subset SL(2,\mathbb{Z}[i])} \\
\begin{pmatrix} g_{0,j} + \mathbf{i}g_{1,j} & g_{2,j} + \mathbf{i}g_{3,j} \\ -g_{2,j} + \mathbf{i}g_{3,j} & g_{0,j} - \mathbf{i}g_{1,j} \end{pmatrix} & \rightarrow & \begin{pmatrix} g_{0,j} + ig_{1,j} & g_{2,j} + ig_{3,j} \\ -g_{2,j} + ig_{3,j} & g_{0,j} - ig_{1,j} \end{pmatrix}
\end{array}$$

# *Collisions pour la fonction LPS [TZ08]*

- Idée de Tillich et Zémor : **relever le problème de représentation** de $PSL(2,\mathbb{F}_p)$ vers $\Omega \subset SL(2,\mathbb{Z}[i])$:

$$\xleftarrow[\text{mod } p]{}$$

$$\begin{array}{rcl}
\mathbf{i}^2 = -1 & \rightarrow & i^2 = -1 \\
\mathbb{F}_p & \rightarrow & \mathbb{Z}[i] \\
PSL(2,\mathbb{F}_p) & \rightarrow & \Omega \subset SL(2,\mathbb{Z}[i]) \\
\begin{pmatrix} g_{0,j} + \mathbf{i}g_{1,j} & g_{2,j} + \mathbf{i}g_{3,j} \\ -g_{2,j} + \mathbf{i}g_{3,j} & g_{0,j} - \mathbf{i}g_{1,j} \end{pmatrix} & \rightarrow & \begin{pmatrix} g_{0,j} + ig_{1,j} & g_{2,j} + ig_{3,j} \\ -g_{2,j} + ig_{3,j} & g_{0,j} - ig_{1,j} \end{pmatrix} \\
\lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \in PSL(2,\mathbb{F}_p) & \rightarrow & \begin{pmatrix} a + bi & c + di \\ -c + di & a - bi \end{pmatrix} \in \Omega
\end{array}$$

# *Ensemble $\Omega$, relevé de $G$*

- Propriétés nécessaires pour $\Omega$:
  - $\Omega \subset SL(2, \mathbb{Z}[i])$
  - La plupart des $m \in \Omega$ ont une factorisation unique par rapport aux relevés des générateurs
  - Cette factorisation se calcule facilement
  - Factorisation dans $PSL(2, \mathbb{F}_p)$ déduite par réduction modulo $p$

# *Ensemble $\Omega$, relevé de $G$*

- Propriétés nécessaires pour $\Omega$:
  - $\Omega \subset SL(2, \mathbb{Z}[i])$
  - La plupart des $m \in \Omega$ ont une factorisation unique par rapport aux relevés des générateurs
  - Cette factorisation se calcule facilement
  - Factorisation dans $PSL(2, \mathbb{F}_p)$ déduite par réduction modulo $p$

- Pour l'ensemble $\Omega$ choisi par [TZ08], trouver $m \in \Omega$ revient à trouver $\lambda, w, x, y, z, e \in \mathbb{Z}$ satisfaisant

$$(\lambda + wp)^2 + 4(xp)^2 + 4(yp)^2 + 4(zp)^2 = l^e$$

# *Ensemble $\Omega$, relevé de $G$*

- Propriétés nécessaires pour $\Omega$:
  - $\Omega \subset SL(2, \mathbb{Z}[i])$
  - La plupart des $m \in \Omega$ ont une factorisation unique par rapport aux relevés des générateurs
  - Cette factorisation se calcule facilement
  - Factorisation dans $PSL(2, \mathbb{F}_p)$ déduite par réduction modulo $p$
- Pour l'ensemble $\Omega$ choisi par [TZ08], trouver $m \in \Omega$ revient à trouver $\lambda, w, x, y, z, e \in \mathbb{Z}$ satisfaisant

$$(\lambda + wp)^2 + 4(xp)^2 + 4(yp)^2 + 4(zp)^2 = l^e$$

Fixer $\lambda + wp$, ...

# *Préimages pour la fonction LPS[PLQ08]*

- Avec la **même stratégie de relèvement**, trouver une préimage d'une matrice $M = \begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix} = \begin{pmatrix} A+B\mathbf{i} & C+D\mathbf{i} \\ -C+D\mathbf{i} & A-B\mathbf{i} \end{pmatrix}$ revient à résoudre
  $(A\lambda + wp)^2 + (B\lambda + xp)^2 + (C\lambda + yp)^2 + (D\lambda + zp)^2 = l^{2k}$

# *Préimages pour la fonction LPS[PLQ08]*

- Avec la **même stratégie de relèvement**, trouver une préimage d'une matrice $M = \begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix} = \begin{pmatrix} A+B\mathbf{i} & C+D\mathbf{i} \\ -C+D\mathbf{i} & A-B\mathbf{i} \end{pmatrix}$ revient à résoudre
  $(A\lambda + wp)^2 + (B\lambda + xp)^2 + (C\lambda + yp)^2 + (D\lambda + zp)^2 = l^{2k}$
- L'extension triviale ne marche pas:
  - Fixer $A\lambda + wp$ pour satisfaire l'équation modulo $p$...
  - ... ne permet pas de simplifier par $p^2$ à cause du terme $2p(wA + xB + yC + zD)\lambda$.
  - Dans l'équation en $x, y, z$ résultante, les coefficients de degré 2 sont très grands (au moins $p$)...
  - ... donc très peu probable d'avoir une solution.

# *Preimages for LPS Hash [PLQ08]*

- Aperçu de notre solution:
  - Solution du problème de préimage pour des *matrices diagonales*
    $(A\lambda + wp)^2 + (B\lambda + xp)^2 + (yp)^2 + (zp)^2 = l^{2k}$
  - Décomposition de toute matrice comme un produit de *matrices diagonales* et des *générateurs*
    $\begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix} = \lambda \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \beta_1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \beta_2 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix}$

# *Preimages for LPS Hash [PLQ08]*

- Aperçu de notre solution:
  - Solution du problème de préimage pour des *matrices diagonales*
    $(A\lambda + wp)^2 + (B\lambda + xp)^2 + (yp)^2 + (zp)^2 = l^{2k}$
  - Décomposition de toute matrice comme un produit de *matrices diagonales* et des *générateurs*
    $\begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix} = \lambda \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \beta_1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \beta_2 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix}$
- Détails en [PLQ08] ou Ch.6

# *Cryptanalyse de la fonction de Morgenstern [PLQ08]*

- Graphes LPS pour des premiers impairs $l$
  Graphes de Morgenstern pour $l^k$, y compris $l = 2$ [M1994]
  Pour efficacité, on prend $l = 2$ [PLQ07]

# *Cryptanalyse de la fonction de Morgenstern [PLQ08]*

- Graphes LPS pour des premiers impairs $l$
  Graphes de Morgenstern pour $l^k$, y compris $l = 2$ [M1994]
  Pour efficacité, on prend $l = 2$ [PLQ07]
- Attaque par relèvement de $SL(2, \mathbb{F}_{2^n})$ vers $\Omega \in SL(2, \mathbb{A})$ où $\mathbb{A} = \mathbb{F}_2[x, y]/(y^2 + y + 1)$
- Equations $\neq$, mais solutions par les mêmes techniques étendues à des polynômes

# *Cryptanalyse de la fonction de Morgenstern [PLQ08]*

- Graphes LPS pour des premiers impairs $l$
  Graphes de Morgenstern pour $l^k$, y compris $l = 2$ [M1994]
  Pour efficacité, on prend $l = 2$ [PLQ07]
- Attaque par relèvement de $SL(2, \mathbb{F}_{2^n})$ vers $\Omega \in SL(2, \mathbb{A})$ où $\mathbb{A} = \mathbb{F}_2[x, y]/(y^2 + y + 1)$
- Equations $\neq$, mais solutions par les mêmes techniques étendues à des polynômes
- Détails en [PLQ08] ou Ch.6

# *Fonction de Zémor-Tillich (ZT)*

- Utilise le graphe de Cayley défini par $G = SL(2, \mathbb{F}_{2^n})$, $v_0 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ et

$$S = \left\{ s_0 = \left(\begin{smallmatrix} X & 1 \\ 1 & 0 \end{smallmatrix}\right), s_1 = \left(\begin{smallmatrix} X & X+1 \\ 1 & 1 \end{smallmatrix}\right) \right\}$$

# *Fonction de Zémor-Tillich (ZT)*

- Utilise le graphe de Cayley défini par $G = SL(2, \mathbb{F}_{2^n})$, $v_0 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ et

$$S = \{s_0 = \left(\begin{smallmatrix} X & 1 \\ 1 & 0 \end{smallmatrix}\right), s_1 = \left(\begin{smallmatrix} X & X+1 \\ 1 & 1 \end{smallmatrix}\right)\}$$

- Résultats partiels de cryptanalyse existants [CP94,G96,AK98,SGGB00]

# Fonction de Zémor-Tillich (ZT)

- Utilise le graphe de Cayley défini par $G = SL(2, \mathbb{F}_{2^n})$, $v_0 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ et
$$S = \left\{ s_0 = \left(\begin{smallmatrix} X & 1 \\ 1 & 0 \end{smallmatrix}\right), s_1 = \left(\begin{smallmatrix} X & X+1 \\ 1 & 1 \end{smallmatrix}\right) \right\}$$
- Résultats partiels de cryptanalyse existants [CP94,G96,AK98,SGGB00]
- Attaques **génériques** pour collision et preimage (utilisant les sous-groupes de $G$) en temps $2^{n/2}$ (au lieu de $2^{3n/2}$ et $2^{3n}$ pour les anniversaires et l'exhaustive) [PQTZ09]

# *Fonction de Zémor-Tillich (ZT)*

- Utilise le graphe de Cayley défini par $G = SL(2, \mathbb{F}_{2^n})$, $v_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ et
$$S = \left\{ s_0 = \begin{pmatrix} X & 1 \\ 1 & 0 \end{pmatrix}, s_1 = \begin{pmatrix} X & X+1 \\ 1 & 1 \end{pmatrix} \right\}$$
- Résultats partiels de cryptanalyse existants [CP94,G96,AK98,SGGB00]
- Attaques **génériques** pour collision et preimage (utilisant les sous-groupes de $G$) en temps $2^{n/2}$ (au lieu de $2^{3n/2}$ et $2^{3n}$ pour les anniversaires et l'exhaustive) [PQTZ09]
- Comments extraire les bits qui sont sûrs ?

# *Zémor-Tillich vectoriel et projectif*

- ZT vectoriel: [PQTZ09]
  - Pour un vecteur initial $\begin{pmatrix} a_0 & b_0 \end{pmatrix}$ partie de la clé,

$$H_{ZT}^{vec}(m) = \begin{pmatrix} a_0 & b_0 \end{pmatrix} H_{ZT}(m)$$

  - Aussi sûre que la fonction initiale

# Zémor-Tillich vectoriel et projectif

- ZT vectoriel: [PQTZ09]
  - Pour un vecteur initial $\begin{pmatrix} a_0 & b_0 \end{pmatrix}$ partie de la clé,

$$H_{ZT}^{vec}(m) = \begin{pmatrix} a_0 & b_0 \end{pmatrix} H_{ZT}(m)$$

  - Aussi sûre que la fonction initiale

- ZT projective: [PQTZ09]
  - Pour un vecteur initial $\begin{pmatrix} a_0 & b_0 \end{pmatrix}$ partie de la clé, renvoie le *point projectif* $[a : b]$ si ZT vectoriel renvoie $\begin{pmatrix} a & b \end{pmatrix}$
  - "Presque" aussi sûre que la version vectorielle

# *Zémor-Tillich vectoriel et projectif*

- (Presque) aussi sûres que la fonction initiale
- La sortie est plus courte: $\approx 3n$ bits $\longrightarrow \approx 2n$ and $\approx n$ bits
- On n'a gardé que la partie "dure" de la recherche de collisions

# *Zémor-Tillich vectoriel et projectif*

- (Presque) aussi sûres que la fonction initiale
- La sortie est plus courte: $\approx 3n$ bits $\rightarrow \approx 2n$ and $\approx n$ bits
- On n'a gardé que la partie "dure" de la recherche de collisions
- Egalement plus efficaces:
  - Toujours pour la version vectorielle
  - Sauf pour les petits messages pour la version projective

# *Zémor-Tillich vectoriel et projectif*

- (Presque) aussi sûres que la fonction initiale
- La sortie est plus courte: $\approx 3n$ bits $\longrightarrow \approx 2n$ and $\approx n$ bits
- On n'a gardé que la partie "dure" de la recherche de collisions
- Egalement plus efficaces:
  - Toujours pour la version vectorielle
  - Sauf pour les petits messages pour la version projective
- Briques de base pour ZesT

# *ZesT: une fonction de hachage tous usages basée sur Zémor-Tillich [PVQ08,PdMQTVZ09]*

- **ZT est attirante:** principales propriétés interprétées en termes de graphes et de groupes, parallélisme, pas trop lent...

## *ZesT: une fonction de hachage tous usages basée sur Zémor-Tillich [PVQ08,PdMQTVZ09]*

- **ZT est attirante:** principales propriétés interprétées en termes de graphes et de groupes, parallélisme, pas trop lent...
- **ZT a des problèmes importants:** malléabilité, invertibilité si messages courts, résistances aux collisions et à la préimage suboptimales

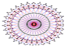
# *ZesT: une fonction de hachage tous usages basée sur Zémor-Tillich [PVQ08,PdMQTVZ09]*

- **ZT est attirante:** principales propriétés interprétées en termes de graphes et de groupes, parallélisme, pas trop lent...
- **ZT a des problèmes importants:** malléabilité, invertibilité si messages courts, résistances aux collisions et à la préimage suboptimales
- **ZesT** est Zémor-Tillich avec Encore plus de Securité dedans

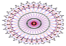
# *ZesT: une fonction de hachage tous usages basée sur Zémor-Tillich [PVQ08,PdMQTVZ09]*

- Utilise les versions vectorielle et projective de ZT

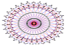
# *ZesT: une fonction de hachage tous usages basée sur Zémor-Tillich [PVQ08,PdMQTVZ09]*

- Utilise les versions vectorielle et projective de ZT
- Résistance aux collisions: même problème que ZT 
- Principales faiblesses de ZT éliminées 

# *ZesT: une fonction de hachage tous usages basée sur Zémor-Tillich [PVQ08,PdMQTVZ09]*

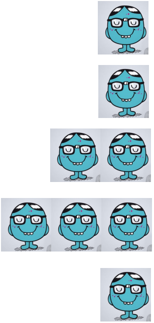
- Utilise les versions vectorielle et projective de ZT
- Résistance aux collisions: même problème que ZT 
- Principales faiblesses de ZT éliminées 
- Implémentations ASIC très légères [dMPQ09]
- Efficacité comparable à SHA sur FPGA [dMPQ09]
- (Pour l'instant) 4 à 10 fois moins rapide que SHA en software

# *ZesT: une fonction de hachage tous usages basée sur Zémor-Tillich [PVQ08,PdMQTVZ09]*

- Utilise les versions vectorielle et projective de ZT
- Résistance aux collisions: même problème que ZT 
- Principales faiblesses de ZT éliminées 
- Implémentations ASIC très légères [dMPQ09]
- Efficacité comparable à SHA sur FPGA [dMPQ09]
- (Pour l'instant) 4 à 10 fois moins rapide que SHA en software
- Parallélisme conservé 

# *Plan de l'exposé*

- Introduction
- Motivations
- Construction et attaques génériques
- Quelques résultats de la thèse
- Conclusion



# Conclusion

- Design simple, clair, élégant
- Securité en terme de propriétés des graphes et des groupes

# *Conclusion*

- Design simple, clair, élégant
- Securité en terme de propriétés des graphes et des groupes
- Aujourd'hui:
  - 1ère fonction de Zémor cassée
  - ZT reste sûre depuis 1994
  - Fonctions de LPS et Morgenstern cassées (et reparées)
  - Fonction de Pizer intacte
  - ZT vectoriel et projectif aussi sûres que ZT

# Conclusion

- Peut être très efficace en software et en hardware
- Parallélisme (fonctions de hachage Cayley)

# *Conclusion*

- Peut être très efficace en software et en hardware
- Parallélisme (fonctions de hachage Cayley)
- Principaux problèmes structurels (malléabilité,...) peuvent être supprimés

# *Conclusion*

- Peut être très efficace en software et en hardware
- Parallélisme (fonctions de hachage Cayley)
- Principaux problèmes structurels (malléabilité,...) peuvent être supprimés
- Besoin de plus d'études, en particulier sur
  - les problèmes mathématiques utilisés
  - la malléabilité des fonctions de hachage

# Conclusion

- Peut être très efficace en software et en hardware
- Parallélisme (fonctions de hachage Cayley)
- Principaux problèmes structurels (malléabilité,...) peuvent être supprimés
- Besoin de plus d'études, en particulier sur
  - les problèmes mathématiques utilisés
  - la malléabilité des fonctions de hachage
- Design très intéressant et prometteur!

# (pré-)Publications liées au sujet de thèse

- **ZesT : an all-purpose hash function based on Zémor-Tillich**
  Christophe Petit, Giacomo de Meulenaer, Jean-Jacques Quisquater, Jean-Pierre Tillich, Nicolas Veyrat-Charvillon and Gilles Zémor
  *Preprint (2009)*
- **Hardware Implementations of a Variant of the Zémor-Tillich Hash Function**
  Giacomo de Meulenaer, Christophe Petit and Jean-Jacques Quisquater
  *Preprint (2009)*
- **Hard and Easy Components of Collision Search in the Zémor-Tillich Hash Function : New Instances and Reduced Variants with Equivalent Security**
  Christophe Petit, Jean-Jacques Quisquater, Jean-Pierre Tillich and Gilles Zémor
  *CT-RSA 2009 - Cryptographer's track at the RSA conference*
- **Full Cryptanalysis of LPS and Morgenstern Hash Functions**
  Christophe Petit, Kristin Lauter, and Jean-Jacques Quisquater
  *SCN 2008 - Sixth Conference on Security and Cryptography for Networks*
- **Efficiency and Pseudo-Randomness of a Variant of Zémor-Tillich Hash Function**
  Christophe Petit, Nicolas Veyrat-Charvillon, and Jean-Jacques Quisquater
  *WIC'2008 - Symposium on Information Theory and Communication in the Bénélux*
  *ISECS'2008 - The 15th IEEE International Conference on Electronics, Circuits and Systems (invited paper)*
- **Cayley Hashes: A Class of Efficient Graph-based Hash Functions**
  Christophe Petit, Kristin Lauter, and Jean-Jacques Quisquater
  *Unpublished (2007)*

# *Autres publications*

- **Fault Attacks on Public Key Elements: Application to DLP based Schemes**
  Chong Hee Kim, Philippe Bulens, Christophe Petit, and Jean-Jacques Quisquater
  *EUROPKI 2008*
- **A Block Cipher based Pseudo Random Number Generator Secure Against Side-Channel Key Recovery**
  Christophe Petit, François-Xavier Standaert, Olivier Pereira, Tal G. Malkin, Moti Yung
  *ASIACCS'08*

# Questions?

# *Il faut fêter ça!*

Un drink vous attend
à la Cafétéria Maxwell
Place du Levant 3