Exploiter l'aléa dans les circuits électroniques pour la sécurité : PUFs et TRNGs

Brice Colombier

brice.colombier@grenoble-inp.fr

Rencontres FMNT
22 octobre 2021

Présentation

Brice Colombier

Enseignant chercheur à Grenoble INP Phelma et au laboratoire TIMA.

Thématique de recherche :

- Sécurité matérielle.

Introduction

Les PUFs [1] et les TRNGs [2] sont des primitives **matérielles** de **sécurité**.

Elles exploitent **l'aléa** présent dans les circuits électroniques.

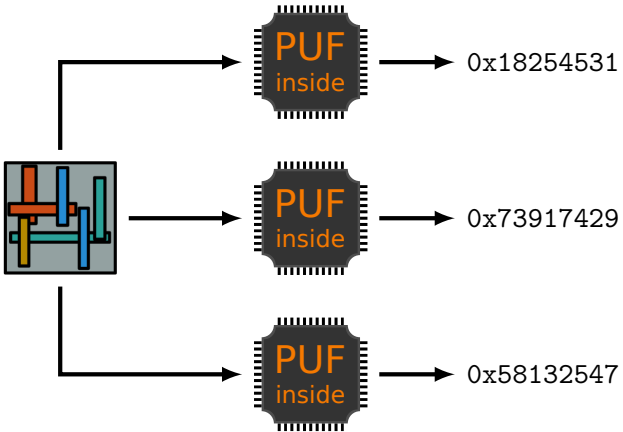
- Une PUF exploite l'aléa **statique**, présent lors de la **fabrication** du circuit.
- Un TRNG exploite l'aléa **dynamique**, présent lors du **fonctionnement** du circuit.

[1] Physical Unclonable Function
[2] True Random Number Generator

Physical Unclonable Function

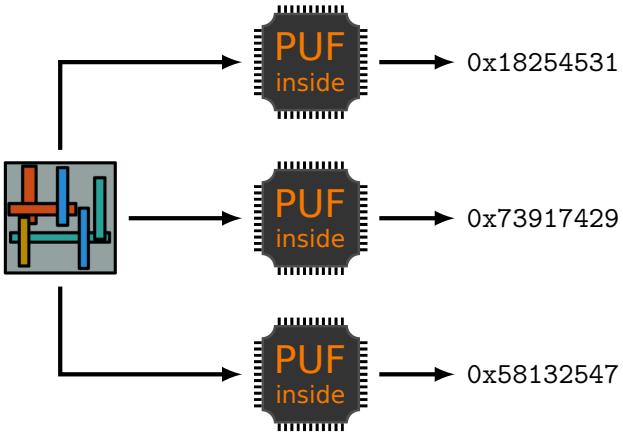
Physical Unclonable Function

Une PUF fournit un **identifiant unique** pour chaque **instance** d'un circuit.



Physical Unclonable Function

Une PUF fournit un **identifiant unique** pour chaque **instance** d'un circuit.

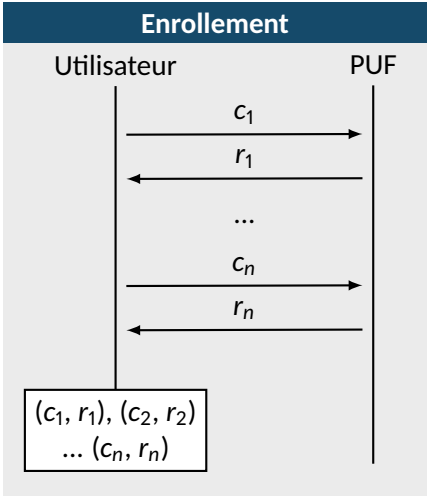


Utilisations possibles

- identification,
- authentification,
- stockage de clés secrètes,
- lutte contre la contrefaçon,

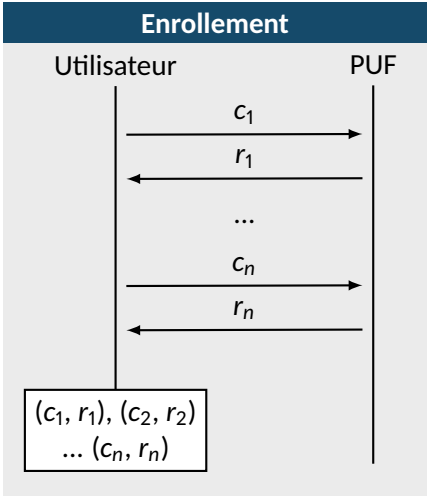
Utilisation

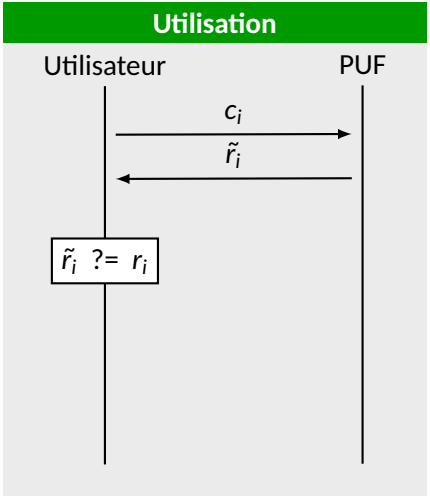
La plupart des PUFs fonctionnent selon un protocole de **challenge / réponse**.



Utilisation

La plupart des PUFs fonctionnent selon un protocole de **challenge / réponse**.





Fonctionnement

Principe de fonctionnement

Comparaison de deux éléments **identiques à la conception** mais **différents après fabrication**.

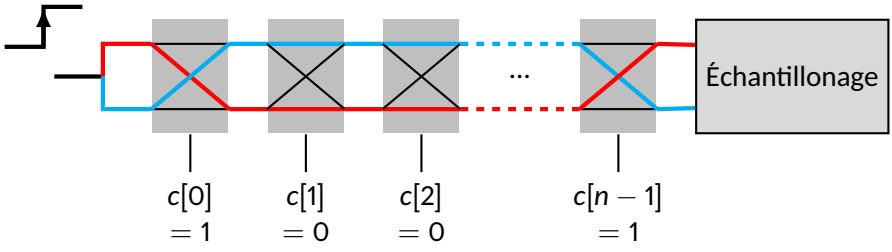
Éléments couramment utilisés :

- lignes,
- oscillateurs en anneau,
- portes logiques,
- cellules mémoire,
- bascules.

PUFs : architectures courantes

Arbiter PUF (2002 [3])

Comparaison du **délai** de deux lignes “identiques”.

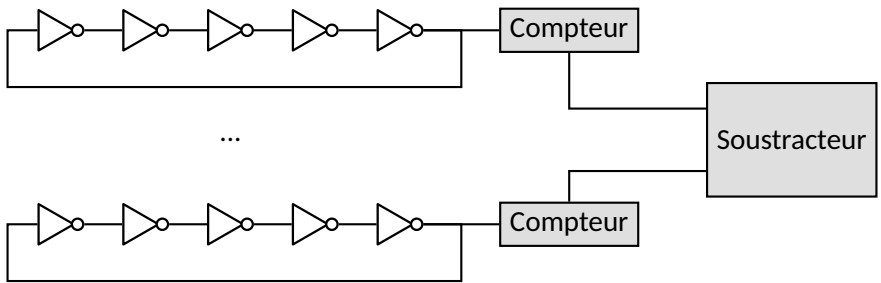


[3] B. Gassend, D. E. Clarke, M. van Dijk, and S. Devadas. “Silicon physical random functions”. In: *ACM Conference on Computer and Communications Security*. ACM, 2002, pp. 148–160.

PUFs : architectures courantes

Ring oscillator PUF (2002 [3])

Comparaison de la **fréquence d'oscillation** de deux oscillateurs en anneau "identiques".

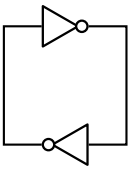


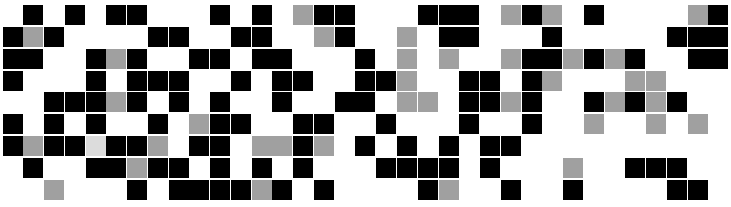
[3] B. Gassend, D. E. Clarke, M. van Dijk, and S. Devadas. "Silicon physical random functions". In: *ACM Conference on Computer and Communications Security*. ACM, 2002, pp. 148–160.

PUFs : architectures courantes

SRAM PUF (2007 [4])

Pour chaque cellule mémoire, l'un des deux inverseurs est plus "fort".
Valeur lue **à la mise sous tension** de la mémoire.





[4] J. Guajardo, S. S. Kumar, G. J. Schrijen, and P. Tuyls. "FPGA Intrinsic PUFs and Their Use for IP Protection". In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Vol. 4727. Springer, 2007, pp. 63–80.

True Random Number Generator

True Random Number Generator

```
10110001000010001111001010011001100111000111001011010110011011111011001000 1
```

Un **T**RNG fournit des nombres aléatoires, utilisés pour des applications de **sécurité** :

- clés secrètes,
- vecteurs d'initialisation,
- masques pour contremesures,

Un TRNG peut aussi fournir une **graine** à un générateur pseudo-aléatoire.

Fonctionnement

Principe de fonctionnement

Extraction du **bruit** présent dans les circuits électroniques

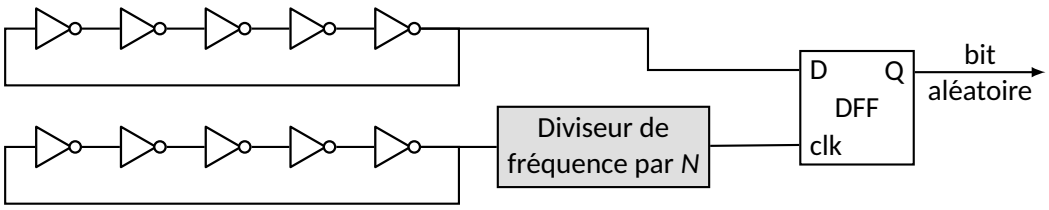
Exemples de phénomènes exploités :

- jitter d'horloge,
- métastabilité,
- délai de transition entre états.

TRNGs : architectures courantes

Ring oscillator TRNG (années 90 [5])

Extraction du signe de la différence de phase après N périodes.

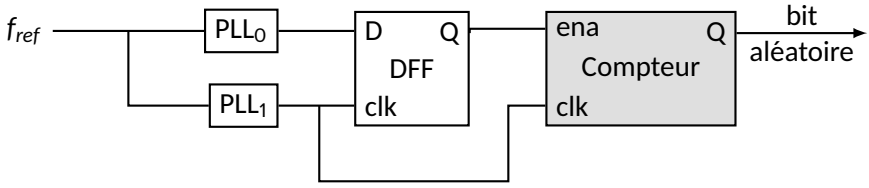


[5] T. E. Tkacik. “A Hardware Random Number Generator”. In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Vol. 2523. Springer, 2002, pp. 450–453.

TRNGs : architectures courantes

PLL TRNG (2004 [6])

Échantillonage cohérent de toute la période puis comptage du nombre de 1.

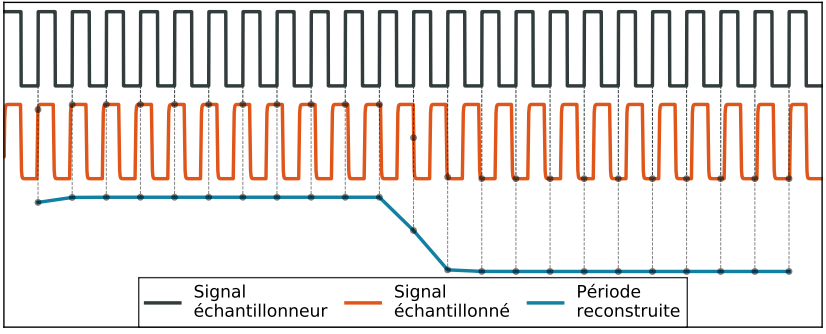


[6] M. Drutarovský, M. Simka, V. Fischer, and F. Celle. “A Simple PLL-Based True Random Number Generator for Embedded Digital Systems”. In: *Computers and Artificial Intelligence* 23.5 (2004), pp. 501–515.

TRNGs : architectures courantes

PLL TRNG (2004 [6])

Échantillonage cohérent de toute la période puis comptage du nombre de 1.

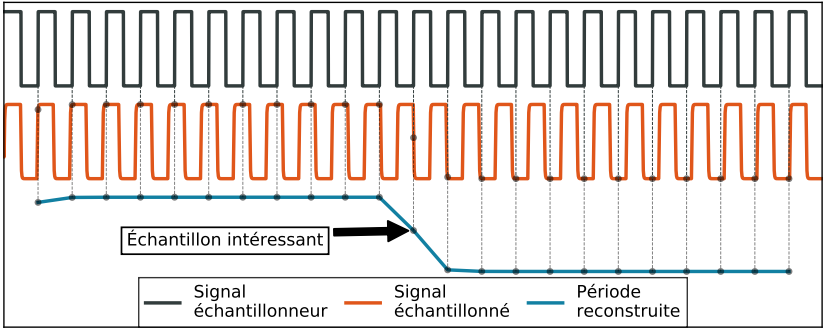


[6] M. Drutarovský, M. Simka, V. Fischer, and F. Celle. “A Simple PLL-Based True Random Number Generator for Embedded Digital Systems”. In: *Computers and Artificial Intelligence* 23.5 (2004), pp. 501–515.

TRNGs : architectures courantes

PLL TRNG (2004 [6])

Échantillonage cohérent de toute la période puis comptage du nombre de 1.

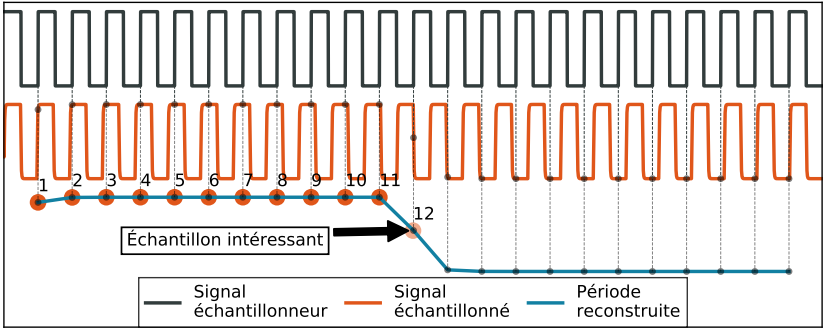


[6] M. Drutarovský, M. Simka, V. Fischer, and F. Celle. “A Simple PLL-Based True Random Number Generator for Embedded Digital Systems”. In: *Computers and Artificial Intelligence* 23.5 (2004), pp. 501–515.

TRNGs : architectures courantes

PLL TRNG (2004 [6])

Échantillonage cohérent de toute la période puis comptage du nombre de 1.

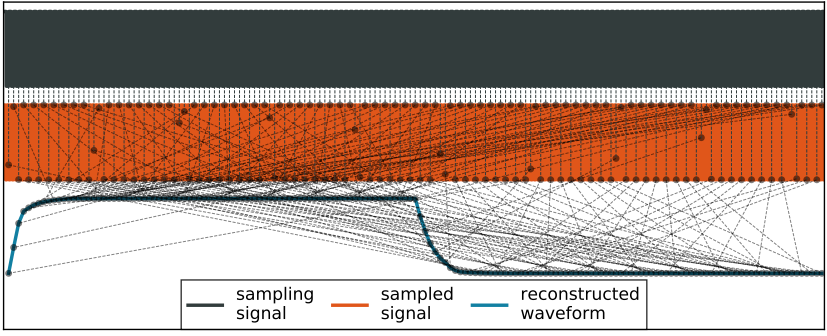


[6] M. Drutarovský, M. Simka, V. Fischer, and F. Celle. “A Simple PLL-Based True Random Number Generator for Embedded Digital Systems”. In: *Computers and Artificial Intelligence* 23.5 (2004), pp. 501–515.

TRNGs : architectures courantes

PLL TRNG (2004 [6])

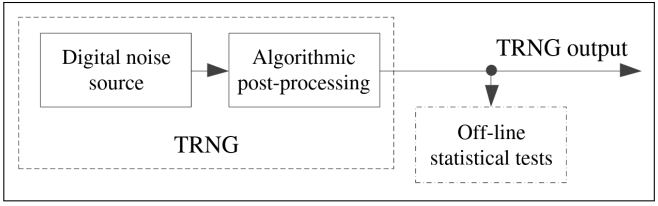
Échantillonage cohérent de toute la période puis comptage du nombre de 1.



[6] M. Drutarovský, M. Simka, V. Fischer, and F. Celle. “A Simple PLL-Based True Random Number Generator for Embedded Digital Systems”. In: *Computers and Artificial Intelligence* 23.5 (2004), pp. 501–515.

Approche historique pour la conception des TRNGs

Approche “historique” :



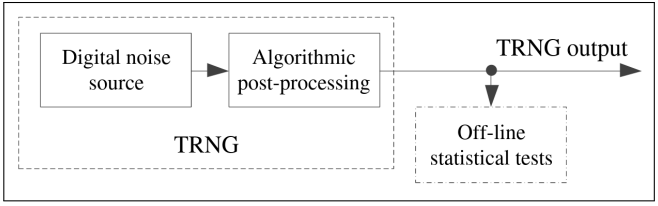
[7]

- Seule la **sortie** du TRNG est testée, par des tests statistiques dédiés
  - FIPS 140-1, NIST SP 800-22, DIEHARD, etc
- Le TRNG est considéré comme une **boîte noire**.

[7] J. Balasch, F. Bernard, V. Fischer, M. Grujic, M. Laban, O. Petura, V. Rozic, G. van Battum, I. Verbauwhede, M. Wakker, and B. Yang. “Design and testing methodologies for true random number generators towards industry certification”. In: *IEEE European Test Symposium*. IEEE, 2018, pp. 1–10.

Approche historique pour la conception des TRNGs

Approche “historique” :



[7]

- Seule la **sortie** du TRNG est testée, par des tests statistiques dédiés
  - FIPS 140-1, NIST SP 800-22, DIEHARD, etc
- Le TRNG est considéré comme une **boîte noire**.

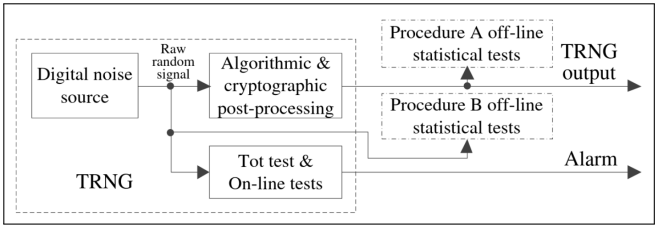
Problème

Des constructions **purement déterministes (sans aucun aléa)** réussissent les tests statistiques.

[7] J. Balasch, F. Bernard, V. Fischer, M. Grujic, M. Laban, O. Petura, V. Rozic, G. van Battum, I. Verbauwhede, M. Wakker, and B. Yang. “Design and testing methodologies for true random number generators towards industry certification”. In: *IEEE European Test Symposium*. IEEE, 2018, pp. 1–10.

Approche moderne pour la conception des TRNGs

Approche “moderne” :



[7]

- Les tests s’appliquent au signal aléatoire brut **et** à la sortie,
- Des **tests** sont intégrés et déclenchent une **alarme** si nécessaire,
- Le TRNG est **précisément décrit**,
- Un **modèle stochastique de l’extraction d’entropie** est requis.

[7] J. Balasch, F. Bernard, V. Fischer, M. Grujic, M. Laban, O. Petura, V. Rozic, G. van Battum, I. Verbauwhede, M. Wakker, and B. Yang. “Design and testing methodologies for true random number generators towards industry certification”. In: *IEEE European Test Symposium*. IEEE, 2018, pp. 1–10.

Propriétés attendues

Propriétés attendues

- **Imprédictibilité**,
  - Indentifiants générés par la PUF uniformément répartis,
  - Impossibilité de prédire les prochains nombres aléatoires générés.
- Résistance aux variations de **conditions de fonctionnement**,
  - Identifiants fiables générés par la PUF,
  - Nombres aléatoires générés fiables.
- Résistance aux **manipulations intentionnelles**,
  - Cloner la PUF,
  - Biaiser la sortie du TRNG.
- **Débit** suffisant,
  - Surtout valable pour les TRNGs.

Comment la microélectronique peut aider ?

- meilleure **caractérisation** des blocs élémentaires,
- caractérisation des **bruits statiques et dynamiques**,
- développement de **modèles stochastiques réalistes et complets**,
  - servant de base aux modèles d'extraction d'entropie,
- utilisation de **nouveaux** blocs élémentaires (technologies émergentes).

Conclusion

Conclusion

- Les PUFs et les TRNGs sont des primitives essentielles pour la **sécurité**.
- Les développer requiert un **modèle stochastique de l'extraction d'entropie**.
- La microélectronique fournit le **socle** nécessaire à la construction de ce modèle.

Contact :

- Ioana Vatajelu : ioana.vatajelu@univ-grenoble-alpes.fr
- Brice Colombier : brice.colombier@grenoble-inp.fr

Conclusion

- Les PUFs et les TRNGs sont des primitives essentielles pour la **sécurité**.
- Les développer requiert un **modèle stochastique de l'extraction d'entropie**.
- La microélectronique fournit le **socle** nécessaire à la construction de ce modèle.

Contact :

- Ioana Vatajelu : ioana.vatajelu@univ-grenoble-alpes.fr
- Brice Colombier : brice.colombier@grenoble-inp.fr

— Questions? —