THE IDENTIFICATION METHOD IN MEMRISTOR-BASED PUF WITH THE TIMING VARIATION

PHƯƠNG PHÁP NHẬN DẠNG PUF GIẢ TRONG ĐIỆN TRỞ NHỚ DỰA TRÊN BỘ NHỚ VỚI SỰ THAY ĐỔI THỜI GIAN

Nguyễn Thị Hà Phương, Hoàng Đình Tuyền
Trường Đại học Quảng Bình

***ABSTRACT**: Nowadays, there are numerous researches on PUF (Physically Unclonable Function) and the application of PUF in electronic devices to increase reliability, against device counterfeiting. However, recognizing a faked PUF has not been studied yet. In this paper, we rarried out research on the methods that identity a faked memristor-based PUF, so that we give the best pseudo-PUF identification method.*

***Keyword:** PUF, memristor, memristor - based PUF.*

***TÓM TẮT:** Hiện nay, có rất nhiều nghiên cứu về PUF (chức năng vật lý không thể mở khoá) và ứng dụng của PUF trong các thiết bị điện tử nhằm làm tăng độ tin cậy, chống lại sự làm giả thiết bị. Tuy nhiên, việc nhận dạng một PUF giả vẫn chưa được các nhà nghiên cứu quan tâm đến. Trong bài báo này, chúng tôi nghiên cứu các phương pháp để nhận dạng một PUF trong điện trở nhớ là giả, từ đó đưa ra được phương pháp nhận dạng PUF giả một cách tối ưu.*

***Từ khoá:** Chức năng vật lý không thể mở khoá, điện trở nhớ, chức năng vật lý không thể mở khoá dựa trên điện trở nhớ.*

1. ĐẶT VẤN ĐỀ

Các chức năng vật lý không thể mở khoá (PUF-Physically Unclonable Function) [1,2] trong các cấu trúc vật lý dễ chế tạo nhưng thực tế không thể sao chép, ngay cả bởi các nhà sản xuất chính hãng của chúng. Những yếu tố này được cho là không thể đoán trước, không thể kiểm soát được, và tạo ra chức năng không thể mở khoá.

Sự phụ thuộc giữa thời gian ghi và sự thay đổi quy trình của điện trở nhớ (memristor) cũng đã được nghiên cứu như một ứng cử viên của PUF. Cách tiếp cận dựa trên thời gian ghi (thời gian lập trình) là một giải pháp đơn giản và dễ hiểu, có thể được thực hiện mà không cần tích hợp nhiều chi phí vào một cấu trúc bộ nhớ thông thường.

Nghiên cứu này nhằm mục tiêu vào việc đưa ra phương pháp phù hợp và tối ưu để nhận dạng một PUF giả và hiện tại chưa có một nghiên cứu nào được đưa ra trước đó. Vì các thiết bị điện tử hiện nay cũng được làm giả rất nhiều, nên chúng tôi đã kết hợp đưa chức năng vật lý PUF vào các thiết bị để nâng cao độ bảo mật và an toàn của thiết bị, bởi lẽ chức năng này rất khó bị gỡ bỏ, tái tạo hay sao chép là điều không thể xảy ra. Chúng tôi đã nghiên cứu phương pháp nhận dạng dựa vào thứ tự lập trình của các điện trở nhớ. Phương pháp đề xuất thu được nhiều mẫu cho kết quả nhận dạng chính xác bằng cách sử dụng các tính năng của các ô ghi nhớ được tổ chức trong một mảng.

Tính hiệu quả của phương pháp lấy

mẫu được xác nhận liên quan đến việc điều khiển điện áp và dòng điện bằng mô hình vật lý của bộ nhớ trong kết cấu thanh ngang.

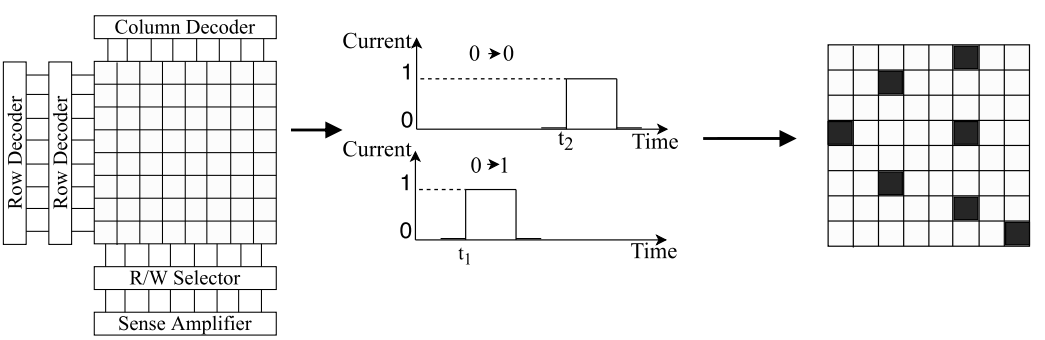
2. MÔ HÌNH ĐỀ XUẤT

Chúng ta cần một nguồn ngẫu nhiên mong muốn trong các ứng dụng mật mã để bảo mật nhằm đảm bảo tính duy nhất được liên kết với một thiết bị. Cho đến nay, việc tạo khóa ngẫu nhiên dựa trên sự biến đổi quy trình của các mạch tích hợp như SRAM và NVM đã được nghiên cứu trong bối cảnh này [3].

Các khoá của mỗi thiết bị là duy nhất, chính vì vậy nó có thể được coi là dấu vân tay để xác định thiết bị cụ thể. Vì sự không ổn định của các ứng dụng trên NVM nên nó thường bị tấn công đối với các ứng dụng bảo mật nâng cao. Tuy nhiên, dung lượng nhỏ hơn và độ tin cậy cao hơn của NVM vẫn hấp dẫn các ứng dụng đơn giản với các yêu cầu bảo mật cơ bản [4].

Hình 1 mô tả quá trình lấy dấu vân tay của thiết bị dựa trên điện trở nhớ. Giá trị logic của các ô điện trở nhớ trong mảng được xác định bởi thời gian truy cập và điện áp cung cấp. Độ dài của xung lập trình và biên độ của dòng lập trình được xác định trước để đảm bảo sự thay đổi trạng thái logic sau khi hoạt động lập trình trong các ứng dụng bộ nhớ. Tuy nhiên, xung lập trình sẽ bị gián đoạn để lấy các giá trị ngẫu nhiên dưới dạng dấu vân tay của thiết bị. Với sự thay đổi trong quá trình sản xuất, trạng thái của các ô điện trở nhớ chỉ bị thay đổi một phần nếu độ dài của xung lập trình hoặc biên độ của xung hiện tại là đủ.

Trong nghiên cứu này, chúng tôi đánh giá giá trị của ô và mảng điện trở nhớ bằng cách sử dụng các mô hình được trình bày trong phần tiếp theo khi xem xét sự thay đổi độ dày của quá trình. Chúng tôi giả định rằng quá trình lập trình có thể bị tạm dừng bằng cách sử dụng bộ điều khiển thời gian trong một phạm vi lỗi nhất định.



Hình 1. Quy trình xử lý của PUF dựa trên điện trở nhớ.
a) Cấu trúc bộ nhớ của điện trở nhớ (trạng thái ban đầu) b) Áp dụng xung lập trình c) Kết quả được tạo ngẫu nhiên cho PUF.

2.1. Mô hình điện trở nhớ với sự biến đổi

Trong tài liệu tham khảo [6,7] mối liên quan giữa thời gian lập trình và nhiệt độ môi trường xung quanh của điện trở nhớ, cấu trúc thanh ngang được khai thác để ước lượng nhiệt độ từ sự biến đổi phụ thuộc nhiệt độ của một điện trở nhớ. Chúng tôi đã sử dụng mô hình vật lý của điện trở nhớ được đề xuất trong [6] để đánh giá hiệu quả phương pháp của chúng tôi. Hoạt động tổng

thể của điện trở nhớ được mô tả qua các công thức trong [7] và thời gian lập trình (hay thời gian ghi) của một ô điện trở nhớ là:

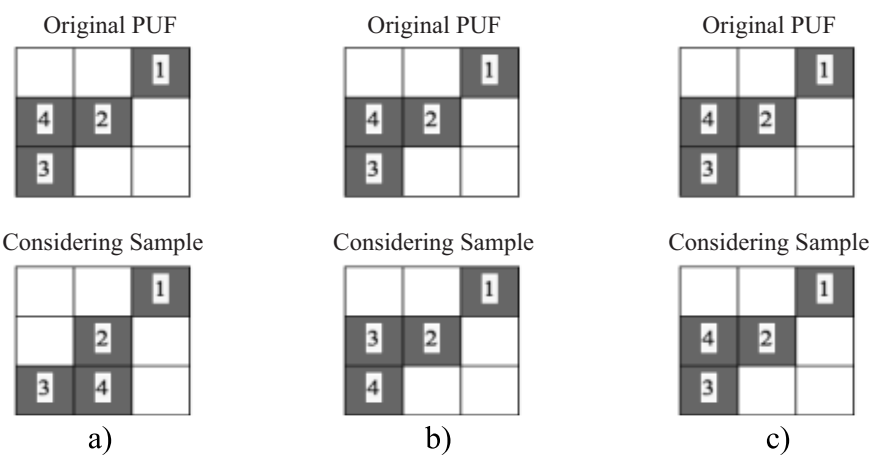
$$t_{write} = \frac{^2D}{\mu_I(T)v_w}\left(\frac{r_1-1}{2}\left(x_0^2 - x_f^2\right) + (r_1 + r_2)\left(x_f - x_0\right)\right) \quad (1)$$

Trong đó D là độ dày của ô điện trở nhớ, $\mu_I(T)$ là tính di chuyển phụ thuộc nhiệt độ, r_1, r_2 phụ thuộc vào R_{off}, R_{on} và $R_{pulldown}$ theo công thức $r_1 = R_{off}/R_{on}, r_2 = R_{pulldown}/R_{on}$.

Thời gian lập trình thường được xây dựng theo sự phân bố Gaussian [7,8] giống như các tham số vật lý khác được sử dụng trong quy trình sản xuất chất bán dẫn.

2.1. Nhận dạng PUF dựa trên điện trở nhớ giả bằng số lượng và vị trí lập trình của các ô điện trở nhớ

Hình 2a diễn tả hai PUF là giống nhau khi số lượng các ô được lập trình giống nhau. Với trường hợp này, chúng tôi chỉ xem xét số lượng của các ô được lập trình, không quan tâm đến vị trí và thứ tự lập trình của các ô. Vì vậy, kết quả nhận dạng các mẫu PUF giả khác với PUF thật là rất ít. Với trường hợp hình 2b, chúng tôi xem xét vị trí của các ô được lập trình (bao gồm cả việc xem xét số lượng của các ô đó). Cả PUF thật và mẫu được xét không những giống nhau về số lượng mà vị trí của các ô được lập trình cũng giống nhau. Và trong trường hợp này PUF giả vẫn có xác suất được nhận dạng khác PUF thật cao hơn trường hợp trước nhưng không đáng kể. Chúng ta có thể lấy một ví dụ như sau để thấy rõ hơn về bản chất của việc nhận dạng sai đó. Ví dụ chúng ta có 4 ô được lập trình tại vị trí 3, 4, 5 và 7, nhưng ô 4 và 7 được lập trình với thứ tự khác nhau. Ô 4 của PUF thật được lập trình thứ 4 và ô 7 được lập trình thứ 3, tuy nhiên ô 4 của mẫu được xét lại được lập trình thứ 3 và ô 7 là thứ 4. Thứ tự lập trình các ô của mẫu được xét khác với mẫu thật, do đó mẫu được xét là giả.



Hình 2. Sự khác nhau giữa PUF thật và giả.
a) Số lượng các ô được lập trình giống nhau b) Vị trí các ô được lập trình giống nhau
c) Thứ tự ghi của các ô được lập trình giống nhau

Khi thực hiện việc ghi một PUF điện trở nhớ tại một thời điểm nhất định, chúng ta sẽ có được số lượng các ô được lập trình, những ô nhớ nào được lập trình và thời gian lập trình của các ô nhớ đó và từ đó biết được thứ tự ghi lần lượt của các ô nhớ [7]. Vậy, nếu tất cả thông tin đó của một mẫu PUF được xem xét để nhận dạng thì độ chính xác sẽ cao hơn rất nhiều so với chỉ xét một trong các thông tin đó. Thế nên, chúng tôi đã đề xuất phương pháp thứ 3 trong Hình 2c là xem xét thứ tự lập trình của các ô. Phương pháp đề xuất xem xét thứ tự lập trình của các ô, bao gồm cả số lượng, vị trí lập trình của các ô điện trở nhớ.

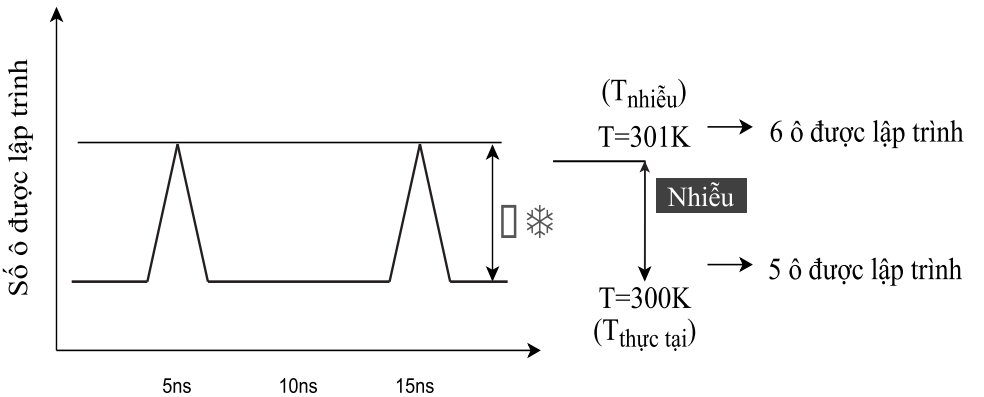
2.1. Nhận dạng PUF dựa trên điện trở nhớ giả bằng thứ tự ghi của các ô điện trở nhớ

Để nhận biết điện trở nhớ là giả hay không (hiện tượng lật bit) bằng cách nhận biết thứ tự ghi của các ô ghi nhớ và số lượng ô được ghi bằng cách sử dụng chức năng không thể mở được về mặt vật lý (PUF). Ví dụ, chúng ta có một PUF điện trở nhớ với số ô được viết tại các khoảng thời gian và ở các nhiệt độ khác nhau là Table 1. Ở các nhiệt độ khác nhau, đồng thời số lượng ô viết cũng khác nhau. Ở nhiệt độ *300K*, số ô được lập trình là *5* ô tại khoảng thời gian *5ns* đầu tiên, *7* ô tại khoảng thời gian *5ns* thứ hai, *9* ô ở *5ns* thứ ba, ... và khi lập trình hết *10.000* ô thì quá trình lập trình của điện trở nhớ PUF đó kết thúc. Nếu nhiệt độ tăng, số lượng ô được lập trình cũng đồng thời tăng lên và thời gian mà tất cả các ô của điện trở nhớ được lập trình sớm hơn.

Bảng 1. Số lượng của các ô được lập trình theo nhiệt độ

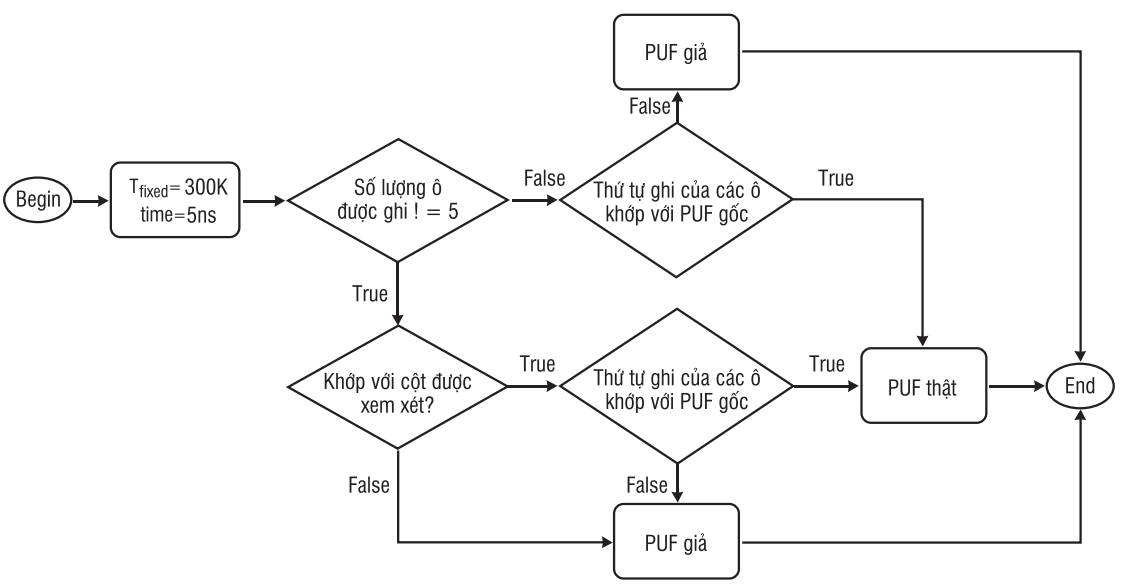
Mốc thời gian	5ns	10ns	15ns	…	*x* ns	Nhiệt độ
Số lượng ô được ghi	5	8	11	…	10.000	T_{fixed}=300K
	6	9	13	…	10.000	T_{fixed}=301K
	7	12	15	…	10.000	T_{fixed}=302K
	9	13	21	…	10.000	T_{fixed}=303K
	…	…	…	…	10.000	…

Tuy nhiên, đó là trong môi trường thực tế. Trên thực tế, khi nhiễu môi trường xảy ra ($T_{thực\ tại} = T_{nhiễu} + \Delta T$). ΔT ảnh hưởng đến các ô ghi nhớ và thay đổi thời gian ghi của mỗi ô (Hình 3). Tại $T_{nhiễu} = 300K$, khoảng thời gian 5ns, số ô được lập trình phải là 5, nhưng trên thực tế, số ô đã thay đổi là 6 và bằng số ô đã thay đổi tại $T = 301K$ ($T_{thực\ tại}$). Trong trường hợp này, điều đó có thể là một chút lật ngược về PUF thật hoặc PUF được sử dụng trong thử nghiệm là giả mạo.



Hình 3. Ví dụ về sự thay đổi của các ô được lập trình vì nhiễu

Phương pháp nhận dạng PUF dựa trên điện trở nhớ (memristor based PUF) bằng vị trí và thứ tự của các ô điện trở nhớ sẽ khắc phục được việc nhận dạng sai do nhiễu gây ra. Chúng ta có sơ đồ để nhận dạng như Hình 4. Với môi trường ban đầu, chúng ta có một điện trở nhớ PUF và thực hiện thao tác lập trình tại $T_{nhiễu} = 300K$ và tại thời điểm *5ns*, chương trình sẽ bị tạm dừng để xem xét số lượng ô được lập trình. Trong môi trường lí tưởng, số ô là *5* ô. Trên thực tế, số ô mà chúng tôi nhận được là khác *5*. Vì vậy, chúng tôi tiếp tục xem xét liệu rằng số ô đó có thuộc một trong các giá trị trong cột *5ns* hay không (Table 1). Nếu nó không thuộc thì PUF đó là giả mạo, ngược lại chúng tôi xem xét thứ tự của các ô được lập trình như thế nào. Nếu thứ tự đó tương tự với thứ tự của các ô đã lập trình của PUF dựa trên điện trở nhớ gốc, thì số lượng các ô đã lập trình của mẫu PUF dựa trên điện trở nhớ sẽ bị thay đổi do lật bit (bit-flipping). Ngược lại, mẫu PUF dựa trên điện trở nhớ, đó đó đã bị làm giả.



Hình 4. Sơ đồ thuật toán của chương trình

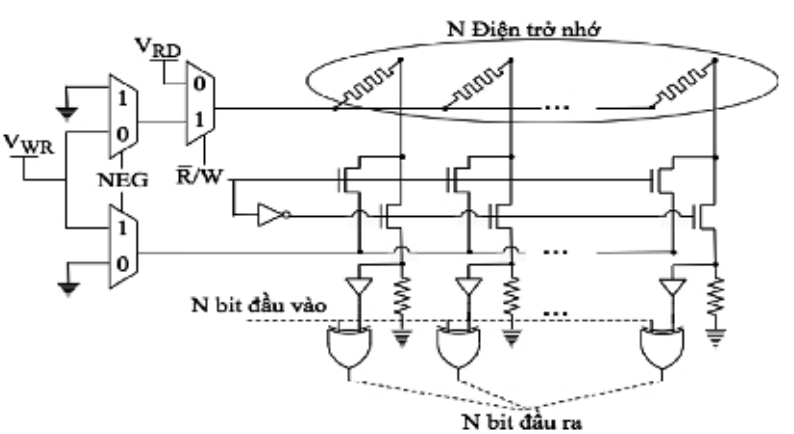
3. KẾT QUẢ THỰC NGHIỆM

3.1. Cơ sở dữ liệu

Thử nghiệm với 10000 mẫu PUF dựa trên điện trở nhớ, mỗi điện trở nhớ có 100x100 ô. Chúng tôi đã triển khai để xử lý các mẫu PUF dựa trên điện trở nhớ với công thức 1 để tính thời gian được lập trình của mỗi ô điện trở nhớ [6] và các tham số, biến trong Bảng 2 để thực hiện việc tạo ra các mẫu PUF ngẫu nhiên và tính toán thời gian ghi. Ngoài ra, chúng tôi còn sử dụng mô hình điện trở nhớ trong cấu trúc mảng thanh ngang được đề xuất trong [9] với N điện trở nhớ, N bit đầu vào (challenge) và N bit đầu ra (Response) cụ thể mô hình được thể hiện trong Hình 5.

Bảng 2. Các tham số và hằng số của mô hình được đề xuất [6]

Tham số/ biến	Giá trị	Mô tả (đơn vị đo)
α	0.15	Khoảng cách nhảy các ion (nm)
E_A	0.18	Năng lượng hoạt động ion (eV
k_B	8.6173303x10^{-5}	Hằng số Boltzmann (eVK^{-1})
f	10	Tần số nhảy ion (THz)
q_I	2	Cường độ ion
v_w	10	Điện áp ghi (V)
T	300-400	Nhiệt độ (K)
D	10	Độ dày điện trở nhớ (nm)
R_{on}	100	Điện trở bật (Ω)
R_{off}	16000	Điện trở tắt (Ω)
$R_{pulldown}$	1000	Điện trở kéo xuống (Ω)
x_0	0	Trạng thái ban đầu của điện trở nhớ
x_f	1	Trạng thái cuối cùng của điện trở nhớ



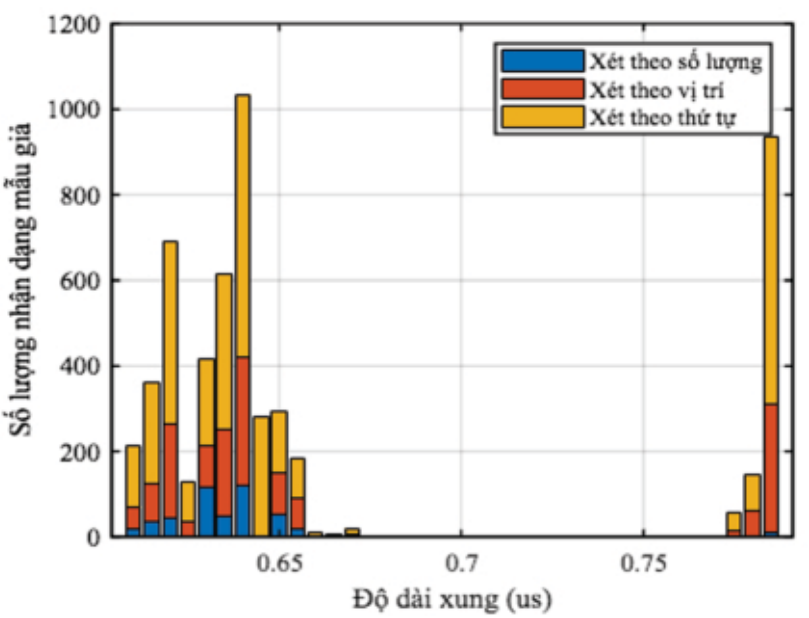
Hình 5. N-bit memrisor-based PUF

3.2. Hiệu suất nhận dạng của các phương pháp

Đầu tiên, với một PUF điện trở nhớ gốc, áp dụng điện áp là *10V* và nhiệt độ là *300K*, chúng tôi đã thực hiện để lập trình PUF điện trở nhớ và tạm dừng hoạt động lập trình tại thời điểm 7.1μs để lấy số lượng các ô đã lập trình và thứ tự lập trình của các ô.

Tiếp theo, chúng tôi đã tạo *10000* PUF điện trở nhớ giả từ cái gốc (được tạo ở bước trước) bằng cách thay đổi độ dày của ô trong phạm vi cho phép. Phạm vi cho phép được tạo bằng cách tăng hoặc giảm độ dày trong phạm vi (*min,sigma_D*) một cách ngẫu nhiên và các ô được chọn cho sự thay đổi ngẫu nhiên đó. Phạm vi (*min,sigma_D*) của độ dày được giới hạn trong phân phối Gaussian $D1 = normrnd\ (mu_D, sigma_D, \#row, \#colum)$.. Số lượng ô thay đổi là *70%* số ô của điện trở nhớ.

Cuối cùng, chúng tôi thực hiện việc tính hiệu suất số mẫu PUF dựa trên điện trở nhớ khác với PUF dựa trên điện trở nhớ gốc với tổng số mẫu được thực nghiệm tại các trường hợp được xét. Quan sát Hình 6 chúng ta thấy được hiệu suất nhận dạng các mẫu giống nhau dựa trên thứ tự lập trình của các ô điện trở nhớ là cao nhất, trường hợp chỉ dựa trên số lượng lập trình của các ô là thấp nhất. Chúng tôi đã thực hiện thí nghiệm tại các độ dài xung khác nhau để thấy rõ hơn độ tin cậy của phương pháp đề xuất. Độ dài xung xa hơn so với 7.1μs thì độ nhận dạng càng thấy được rõ hơn, đặc biệt với trường hợp chỉ xét số lượng hay vị trí của các ô lập trình, số lượng nhận dạng các mẫu khác nhau có khi chỉ bằng 0. Xác suất nhận dạng ra mẫu giả của các trường hợp xét theo số lượng, vị trí và thứ tự của ô được lập trình lần lượt là *0.13%*, *0.43%* và *0.94%*. Phương pháp đề xuất có hiệu suất tăng gấp 2.2 lần so với phươn g pháp chỉ xét theo vị trí và 7.2 lần so với phương pháp chỉ xét theo số lượng.



Hình 6. Số lượng nhận dạng các mẫu giống nhau theo số lượng, vị trí và thứ tự lập trình của các ô điện trở nhớ

4. KẾT LUẬN

Chúng tôi đã nghiên cứu về PUF dựa trên điện trở nhớ, đồng thời xem xét những bất cập trong việc nhận dạng PUF giả dựa và các phương pháp truyền thống. Từ đó, đưa ra được phương pháp nhận dạng bằng cách dựa vào thứ tự lập trình của các ô điện trở nhớ. Hiệu quả của phương pháp đề xuất lên đến 0.94% gấp 7.2 lần so với phương pháp truyền thống. Bên cạnh đó, kết quả của thử nghiệm này cho thấy xác suất trùng khớp chính xác giữa những cái gốc và cái giả được dự kiến là thấp đáng kể với độ dài bit đủ dài. Tuy nhiên, khi chúng tôi xem xét ảnh hưởng của sự thay đổi thời gian trong quá trình đăng ký, xác suất nhận dạng sai cho biết giá trị cao hơn đáng kể trong hệ thống thực [5].

Trong tương lai, chúng tôi sẽ tìm hiểu các phương pháp nâng cao việc nhận dạng một PUF giả, làm tăng hiệu suất nhận dạng bằng cách phân chia khung thời gian nhỏ hơn hay thực hiện ghi ngược để xem xét lại các mẫu nhận dạng sai.

TÀI LIỆU THAM KHẢO

[1] J. Mathew, R.S. Chakraborty, D. P. Y. Yang, và D. K. Pradhan (2015), *A novel memristor based physically unclonable function*, Tạp chí Integration, the VLSI, tập 51, trang 37-45.

[2] R. Maes và I. Verbauwhede (2010), *Physically unclonable functions: A study on the state of the art and future research directions*, Tạp chí Towards Hardware-Intrinsic Security, Springer, Berlin, Heidelberg, trang 3-37.

[3] S. Eiroa, J. Castro, M. C. Martinez-Rodriguez, E. Tena, P. Brox, và I. Baturone (2012), *Reducing bit flipping problems in SRAM physical unclonable functions for chip identification*, Kỷ yếu hội nghị IEEE Electronics, Circuits and Systems (ICECS), 2012 19th IEEE International Conference, trang 392-395.

[4] W. Che, J. Plusquellic, và S. Bhunia (2014), *A non-volatile memory based physically unclonable function without helper data*, Ký yếu hội nghị Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design}, trang 148-153.

[5] M. Yu và S. Devadas (2010), *Secure and robust error correction for physical unclonable functions*, Tạp chí IEEE Design Test of Computers, tập 27, số 1, trang 48-65.

[6] C. Merkel (2011), *Thermal profiling in CMOS/memristor hybrid architectures*, NXB Rochester Institute of Technology.

[7] T.N. Nguyen và D. Shin (2018), *Statistical Memristor-Based Temperature Sensors without Analog-to-Digital Conversion*, Kỷ yếu hội nghị 2018 IEEE 7th Non-Volatile Memory Systems and Applications Symposium (NVMSA)}, trang 99-104.

[8] H. H. Li và M. Hu (2010), *Compact model of memristors and its application in computing systems*, Kỷ yếu hội nghị Proceedings of the Conference on Design, Automation and Test in Europe, European Design and Automation Association, trang 673-678.

[9] G. S. Rose, N. McDonald, L. K. Yan, và B. Wysocki (2013), *A write-time based memristive PUF for hardware security applications*, Kỷ yếu hội nghị IEEE/ACM International Conference 2013, trang 830-833.

Liên hệ:
TS. Nguyễn Thị Hà Phương
Khoa Kỹ thuật - Công nghệ thông tin, Trường Đại học Quảng Bình
Địa chỉ: 312 Lý Thường Kiệt, Đồng Hới, Quảng Bình
Email: nguyenphuong18285@gmail.com

Ngày nhận bài: 18/11/2021
Ngày gửi phản biện: 20/11/2021
Ngày duyệt đăng: 30/01/2022