دانشگاه تهران

پردیس دانشکده های فنی

دانشکده مهندسی برق و کامپیوتر

پیاده‌سازی حملات مبتنی بر یادگیری ماشین برعلیه توابع غیرهمسان فیزیکی

نگارش:

سید ابوالفضل سجادی هزاوه

استاد راهنما: دکتر بیژن علیزاده ملفه

پایان نامه برای دریافت درجه کارشناسی ارشد

در

مهندسی برق گرایش سیستم‌های الکترونیک دیجیتال

شهریور ماه ۱۴۰۰


بسم الله الرحمن الرحيم

دانشگاه تهران

پردیس دانشکده های فنی

دانشکده مهندسی برق و کامپیوتر

پیاده‌سازی حملات مبتنی بر یادگیری ماشین برعلیه توابع غیرهمسان فیزیکی

نگارش:

سید ابوالفضل سجادی هزاوه

استاد راهنما: دکتر بیژن علیزاده ملفه

پایان نامه برای دریافت درجه کارشناسی ارشد

در

مهندسی برق گرایش سیستم‌های الکترونیک دیجیتال

شهریور ماه ۱۴۰۰

بسمه تعالی

گواهی دفاع از پایان نامه کارشناسی ارشد

هیأت داوران پایان نامه کارشناسی ارشد آقای/خانم: سیدابوالفضل سجادی هزاوه در رشته مهندسی برق /کامپیوتر، گرایش : سیستمهای دیجیتال

را، با عنوان پیاده سازي حملات مبتني بر يادگيري ماشين بر عليه توابع غير همسان فيزيكي

در تاریخ ۱۴۰۰/۰۶/۲۷ با درجه عالی ارزیابی نمود.

مشخصات هیات داوران	نام و نام خانوادگی	مرتبه دانشگاهی	دانشگاه یا موسسه	امضاء
- استاد راهنمای اول - استاد راهنمای دوم (حسب مورد)	دکتر بیژن علیزاده ملفه	دانشیار	تهران	
- استاد مشاور - استاد مشاور دوم (حسب مورد)	--	--	--	
- استاد مدعو خارجی	دکتر علی جهانیان	دانشیار	شهید بهشتی	
- داور و نماینده تحصیلات تکمیلی دانشکده	دکتر مهدي كمال	دانشیار	تهران	

تذکر:این برگ پس از تکمیل توسط هیات داوران در نخستین صفحه پایان نامه درج می گردد.

معاونت تحصیلات تکمیلی
پردیس دانشکده های فنی
تهران

تحصیلات تکمیلی

تعهدنامه اصالت اثر

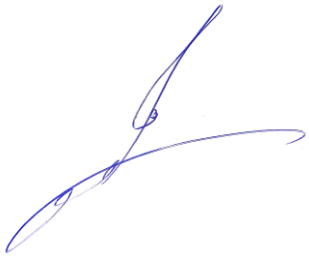
باسمه‌تعالی

اینجانب **سید ابوالفضل سجادی هزاوه** تائید می‌کنم که مطالب مندرج در این پایان‌نامه حاصل کار پژوهشی اینجانب است و به دستاوردهای پژوهشی دیگران که در این نوشته از آن‌ها استفاده شده است مطابق مقررات ارجاع گردیده است. این پایان‌نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نشده است.

کلیه حقوق مادی و معنوی این اثر متعلق به دانشکده فنی دانشگاه تهران می‌باشد.

نام و نام خانوادگی دانشجو : سید ابوالفضل سجادی هزاوه

امضای دانشجو :



این پایان نامه را تقدیم میکنم به

بزرگ ترین و ارزشمند ترین آموزگاران زندگی ام

که همواره برایم تکیه گاه امن و مطمئنی بودند.

پدر و مادر عزیزم

هزاران بار سپاس.

تشکر و قدردانی:

از تمام عزیزانی که من را در مسیر این پژوهش یاری دادند متشکرم.

- با تشکر و سپاس از استاد دانشمند و پرمایه‌ام جناب آقای دکتر علیــزاده کــه از محضــر پرفیض تدریسشان ، بهره‌ها برده‌ام.

- از دوستان گرامی آزمایشگاه "طراحی، درستی سنجی و عیب‌یابی سیستم‌های نهفتــه" و دیگر عزیزانی که با ایجاد محیطی گرم، من را در این طرح یاری نمودند سپاسگزارم.

چکیده

صنعت اینترنت‌اشیا با چالش‌های فراوانی روبه‌رو است که یکی از مهمترین چالش امنیت در ارتباطات است، همچنین یکی از مهمترین عمل‌ها برای برقراری امنیت در ارتباطات، مبحث احرازهویت[1] است. بدین معنا که پیام دریافت شده را چه کسی ارسال کرده است. باتوجه به حمله بازپخش[2] و سایر حملات این حوزه، مهاجم حتی درصورت دستیابی به پیام رمزنگاری شده، می‌تواند پیام نامفهوم دریافت شده را در زمان مناسب ارسال نموده و به اهداف شوم خود برسد، بنابراین حتی پیچیدگی در رمزنگاری داده هم منجر به ایمن شدن شبکه نمی‌شود؛ بنابراین وجود راهکارهای وابسته به داده‌های سیگنال دریافتی، اعم از رمز و یا سایر روش‌ها، نمی‌تواند به تنهایی راهکاری مناسب برای احرازهویت باشد.

روش پیشنهادی جهت احرازهویت استفاده از پارامترهای فیزیکی ارسال کننده است. این پارامترها در فرآیند ساخت به صورت تصادفی بوجود آمده‌اند، بنابراین از کنترل انسان جهت اعمال تغییرات خارج است، این پروتکل‌ها را توابع غیرهمسان فیزیکی[3] می‌نامند. در این مطالعه به بررسی مقاومت برخی از این پروتکل‌ها پرداخته‌شده است. با توجه به مزیت‌ها و مشکلات این مدل‌ها یک مدل جدید به نام SQ-PUF ارائه‌شده که می‌تواند مقاومت خوبی در برابر حملات مبتنی بر یادگیری ماشین از خود نشان دهد. این یک معماری امن و مقیاس‌پذیر است.

در آخر ما مقاومت این مدل را نسبت به حملات یادگیری ماشین بررسی کردیم، مدل ما توانست مقاومت خوبی در برابر حملات یادگیری ماشین از خود نشان دهد و این حملات نتوانستند بیشتر از ۵۲.۵٪ حمله موفق داشته باشند همچنین گزارش‌هایی از یکنواختی و یکتایی داده‌ها و سربار سخت‌افزاری مدل فوق ارائه دادیم.

واژه‌های کلیدی: اینترنت‌اشیا، یادگیری ماشین، احرازهویت، امنیت شبکه، توابع غیرهمسان فیزیکی

---

[1] Authentication
[2] Replay attack
[3] Physical unclonable function

فهرست مطالب

فهرست اشکال

فهرست جداول

فهرست علائم اختصاری

توضیح	علامت
اینترنت‌اشیا	IoT
توابع غیر همسان فیزکی	PUF
یادگیری ماشین	ML
فاصله همینگ	HD
چالش-پاسخ	CRP
آرایه درگاه قابل برنامه‌ریزی	FPGA
تولید کننده عدد تصادفی	RNG
مالتی‌پلکسر	MUX

فصل۱: مقدمه

۱-۱- عنوان تحقیق

پیاده‌سازی حملات[1] مبتنی بر یادگیری ماشین[2] برعلیه توابع غیرهمسان فیزیکی[3].

۱-۲- مسئله تحقیق و تعریف مسئله

در عصر حاضر پیشرفت پروتکل‌های ارتباطاتی[4] در مدارات الکترونیکی و همچنین افزایش استفاده از اینترنت‌اشیا بر اهمیت بحث انتقال داده و امنیت[5] آن، در ارتباطات بین دو دستگاه افزوده است. دلایل اهمیت این موضوع را می‌توان پیشرفت روزافزون و تعداد این دستگاه‌ها برشمرد، به تعداد نزدیک بر ۲۵ میلیارد دستگاه متصل به شبکه، بیشتر از ۵۰ تریلیون گیگابایت داده و ۴ میلیارد انسان متصل به شبکه اشاره کرد. با توجه به گستردگی ارتباطات بین دستگاه‌ها، چالش‌های زیادی برای این ساختارها در پیش رو است، یکی از مهم‌ترین این چالش‌ها، بحث امنیت در انتقال

1 Attack

2 Machine Learning (ML)

3 Physically Unclonable Function (PUF)

4 Communication protocols

5 Security

داده‌ها می‌باشد[2] ,[1].

پیشرفت فناوری‌های مختلف ازجمله شناسایی و ردیابی خودکار، حسگرها، محاسبات نهفته، دسترسی باند گسترده به اینترنت و سرویس‌های توزیع‌شده، ظرفیت ادغام اشیا هوشمند را در رندگی روزمره ما از طریق اینترنت افزایش می‌دهد. همگرایی اینترنت و اشیا هوشمندی که می‌توانند به برقراری ارتباط و تعامل با یکدیگر بپردازند، اینترنت اشیا (IOT) را تعریف می‌کند.

بسیاری از حوزه‌های کاربردی مانند ایمنی عمومی، خودکارسازی منازل، نظارت بر محیط و فرآیندهای صنعتی از مزایای قابل‌توجه اینترنت اشیا بهره می‌برند. با این وجود ادغام اشیای موجود و اینترنت می‌تواند تهدیدهای امنیتی را نیز به همراه داشته باشد که عواقب سنگینی را به‌خصوص برای زیرساخت‌های مهم و حیاتی مانند نیروگاه‌های انرژی و سیستم حمل‌ونقل داشته باشد.

به طورکلی امنیت پروتکل‌های ارتباطی شامل دو قسمت یکپارچگی داده[1] و احرازهویت[2] می‌باشد. جهت حفظ یکپارچگی داده، داده ارسال‌شده با استفاده از طرح‌های رمزنگاری[3]، امن می‌گردد، بدین صورت که اگر مهاجم به کانال ارتباطی[4] دسترسی پیدا کند، داده رمزنگاری شده[5] نامفهومی را مشاهده می‌کند که توانایی برداشتن اطلاعات کمی را داشته باشد. طرح رمزنگاری استفاده شده دارای پیچیدگی هستند، پس می‌توان میزان دریافت اطلاعات درست توسط حمله کننده را تخمین زد. در صورتی که طرح رمزنگاری پیچیده‌تری ارائه شود، می‌توان احتمال جلوی درز کردن اطلاعات را در هنگام حمله گرفت. طراحی الگوریتم رمزنگاری پیچیده، علاوه بر مباحث سنگین ریاضیاتی، نیاز به توان مصرفی و فضای پیاده‌سازی بیشتری در فاز پیاده‌سازی و در نتیجه هزینه سخت‌افزاری بیشتری دارد، البته احتمال حمله موفق و خواندن اطلاعات همچنان وجود دارد.

[1] Data Integrity
[2] Data Authentication
[3] Cryptography Scheme
[4] Communication Channel
[5] Encrypted Data

دومین چالش درباره امنیت در پروتکل‌های ارتباطی بحث احرازهویت است، یعنی تعیین مبدأ داده ارسال‌شده می‌باشد. بحث احرازهویت زمانی مطرح می‌شود که احتمال ارسال اطلاعات توسط مهاجم وجود داشته باش، ازین‌رو گیرنده پیام باید توانایی تشخیص فرستنده اصلی از مهاجم را داشته باشد، در غیر این صورت ممکن است توسط مهاجم مورد حمله قرار گیرد.

الگوریتم‌های رمزنگاری هم نمی‌تواند به تنهایی امنیت را تضمین کند، به عنوان مثال، یک فرد مهاجم داده‌ی رمزنگاری شده‌ای که نامفهوم است را در اختیار دارد که نمی‌تواند آن را رمزگشایی کند، اما نتیجه اجرا شدن آن را می‌داند، یعنی می‌داند در صورتی که این پیام را ارسال کند، کار خاصی را انجام می‌دهد، بنابراین با ارسال کردن این پیام در زمان مناسب می‌تواند به هدف موردنظر برسد. این مثال به‌خوبی می‌تواند نیاز به احرازهویت در امنیت پروتکل‌های ارتباطی حتی در صورت استفاده از طرح رمزنگاری پیچیده و موفق و دقیق را خاطرنشان کند. به این نوع حمله بازپخش[1] گفته می‌شود[3].

امروزه در حوزه امنیت سخت‌افزار مفهومی با عنوان PUF[2] مطرح‌شده است. در واقع مفهوم PUF همانند اثرانگشت انسان است که برای مدارهای الکترونیکی عمل می‌کند. سه دلیل برای این شباهت وجود دارد: ۱- اثرانگشت[3] جزو ویژگی‌های فردی است نه حیاتی، بدین معنی که جزو ویژگی‌هایی مشخص کننده انسان بودن همچون دو دست، دو پا و ... نبوده، بلکه تعیین کننده شخص بوده و هر فردی اثرانگشت منحصر به فرد خود را دارد. بدین معنا است که در جواب سؤال کدام انسان (نه اینکه آیا انسان است یا خیر) مطرح می‌شود. PUF نیز ویژگی‌های منحصر به فرد هر مدار مجتمع[4] را داراست و مشخص کننده مدار مجتمع نیست، بلکه مشخص کننده نوع مدار مجتمع

[1] Replay Attack
[2] Physical Unclonable Function
[3] Fingerprint
[4] Integrated Circuit

است. ۲- اثرانگشت برای هر انسان به‌صورت منحصر به فرد[1] و برای هر انسان با افراد دیگر متفاوت است، PUF نیز برای هر مدار مجتمع با سایر مدارات متفاوت است.۳- اثرانگشت با سایر ویژگی‌های انسان از جمله امضا، اسم، کدملی و ... تفاوت دارد، اثرانگشت جزو ویژگی‌های ذاتی[2] و غیرقابل تغییر می‌باشد، در حالی که کدملی و اسم بعد از تولد انسان تعیین می‌شوند و قابل تغییر هستند، PUF نیز در مراحل ساخت وابسته به اثرات تغییر پارامتر در هنگام فرآیند ساخت[3] می‌باشد، بنابراین یک مشخصه‌ی ذاتی است.

معماری[4] ارائه‌شده برای PUF وابسته به تغییرات پارامترهایی همچون تأخیر مسیرها و مقادیر خازن‌ها و ... در زمان فرآیند ساخت مدارهای مجتمع بوده و با توجه به اینکه تغییرات فرآیندساخت، به‌صورت کاملاً تصادفی تعیین می‌شوند، نحوه عملکرد PUF نیز به‌صورت منحصر به فرد و تصادفی است. به‌طور کلی PUF-ها دو کاربرد مهم دارند که شامل تولید کلید و احرازهویت در مدارات مجتمع بدون نیاز به ذخیره‌سازی آن در حافظه دیجیتال می‌باشد. از این معماری می‌توان جهت جلوگیری از نمونه‌برداری غیرمجاز[5] مدارات مجتمع و همچنین تولید کلید برخط[6] و مقاوم در برابر حملات تهاجمی[7]، بهره گرفت. در صورتی که طراحی مناسبی از PUF-ها ارائه شود، احتمال جعل و نسخه‌برداری بدون اجازه یا به عبارتی حمله موفق به این مدارات به شدت کاهش پیدا خواهد کرد، به‌گونه‌ای که از این معماری با عنوان "بازگرداننده امید به امنیت در الکترونیک" یاد می‌شود.

در این پایانامه ضمن بررسی انواع روش‌های مختلف حمله به PUFهای مقاوم روش‌های حمله دیگری نیز ارائه شده است. سپس با بررسی این روش‌ها و استفاده از مزایا و معایب مدل‌های قبلی

[1] Unique
[2] Inherent
[3] Process Variation
[4] Architecture
[5] Counterfeiting
[6] Online
[7] Tamper Attack

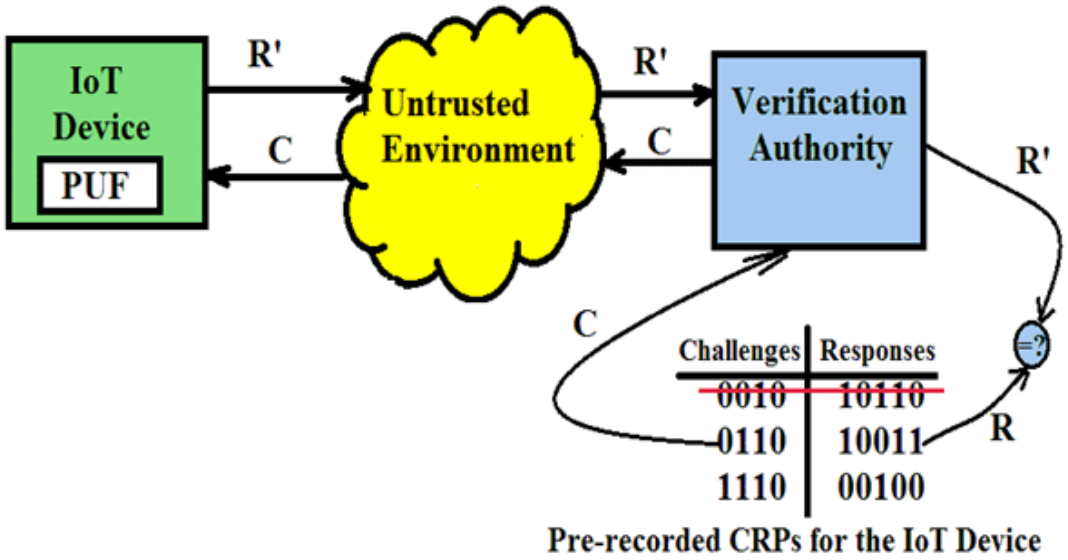
پروتکل جدیدی از PUF ارائه شده که می‌تواند مقاومت خوبی در برابر حملات یادگیری ماشین ارائه شده از خود نشان دهد.

پیاده‌سازی‌های متفاوتی با هدف بهبود امنیت با استفاده از PUF ارائه گردیده است که در فصل دوم به توضیح مفصل‌تر آن پرداخته خواهد شد.

۱-۳- تاریخچه ای از موضوع تحقیق

مقاله [4] برای اولین بار مفهوم PUF را در مدارات مجتمع سیلیکونی مطرح نمود. در این معماری دیگر کلید رمزنگاری در حافظه دیجیتال ذخیره نمی‌شد، پس می‌توان این معماری را در برابر حملات مستقیم و غیرمستقیم مقاوم دانست. از آن به بعد روش‌های گوناگونی برای حمله به این ساختارها و شکستن مقاومت این ساختارها ارائه شد و به طبع آن ساختارهای PUF با توجه به این روش‌ها مقاوم‌تر شدند. در شکل (۱-۱) نمونه‌ای از ارتباط بین یک PUF و پروسه احرازهویت آورده شده است. منظور از حمله به PUF پیدا کردن رابطه بین ورودی و خروجی مدل مورد نظر به وسیله متدهای یادگیری ماشین می‌باشد.

در فصل دوم برخی از مهم‌ترین این ساختارها را معرفی می‌کنیم و در فصل سوم به بررسی بیشتر حملات و ساختارهای مقاوم خواهیم پرداخت.



شکل (۱-۱) شمایی از پروسه احرازهویت توسط PUF [5]

۱-۴- نوآوری، اهمیت و ارزش تحقیق

اینترنت اشیا مفهومی است که انواع مختلف برنامه‌های کاربردی را در کنار یکدیگر جمع‌آوری کرده تا بر اساس همگرایی اشیا هوشمند و اینترنت، ادغامی بین جهان فیزیکی و سایبری ایجاد کند؛ که این برنامه‌های کاربردی می‌تواند از یک دستگاه ساده در یک‌خانه هوشمند تا تجهیزات پیچیده در یک کارخانه‌ی صنعتی را در برگیرد.

با وجود اهداف متفاوت برنامه‌های کاربردی اینترنت اشیا، همه‌ی آن‌ها مشخصات مشترکی را به اشتراک می‌گذارد. به‌طورکلی اینترنت اشیا شامل سه مرحله مجزا شامل مرحله جمع‌آوری، مرحله‌ی انتقال و مرحله‌ی پردازش، مدیریت و مصرف می‌باشد. با توجه به افزایش رشد انواع دستگاه‌های متصل به شبکه اینترنت اشیا، گسترش و بهبود تهدیدهای امنیتی از اهمیت بالایی برخوردار است. اگرچه استفاده از اینترنت اشیا بهره‌وری و کیفیت زندگی مردم را افزایش می‌دهد،

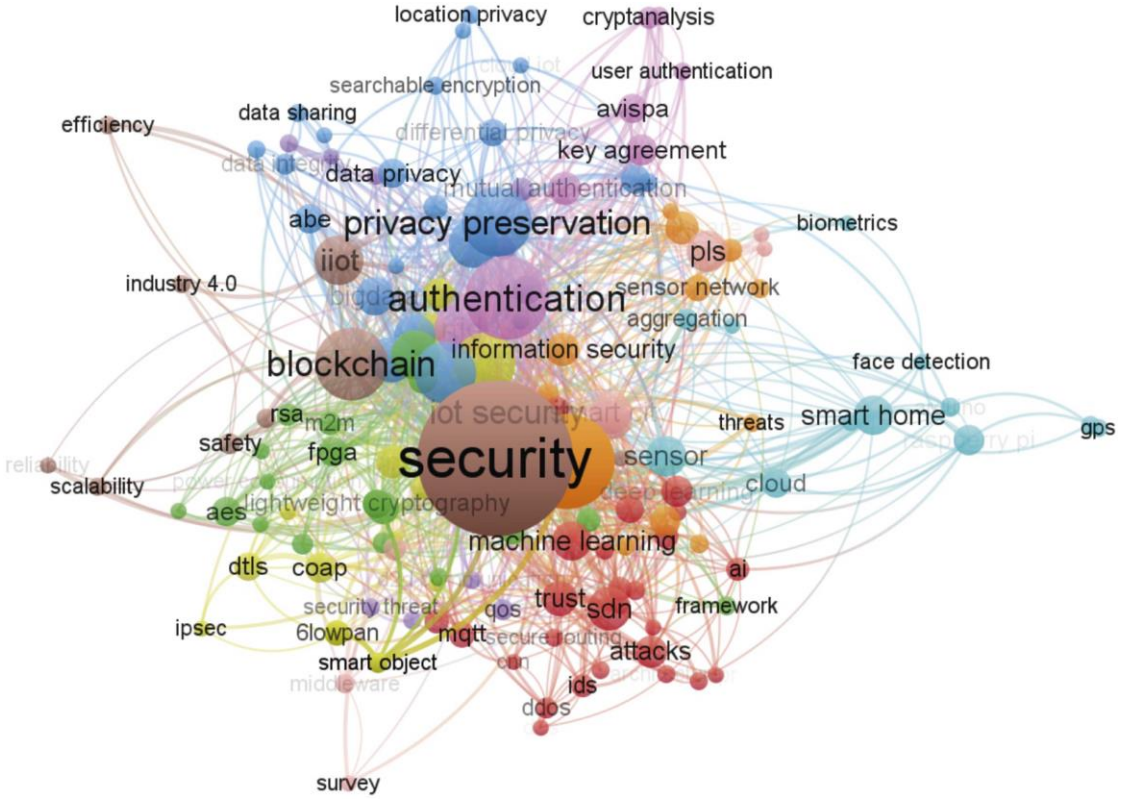
بااین‌وجود مهاجمان ممکن است از این ظرفیت بزرگ اینترنت اشیا برای تهدید امنیت و حریم خصوصی سیستم‌های مختلف استفاده کنند، بنابراین راه‌حل‌های امنیتی برای اینترنت اشیا باید توسعه داده شود.دستگاه‌های اینترنت اشیا به علت عدم رمزگذاری انتقال، رابط وب ناامن، حفاظت ناکافی از نرم‌افزار، بسیار آسیب‌پذیر می‌باشند. برخی برنامه‌های اینترنت اشیا از زیرساخت‌های حساس و خدمات استراتژیک مانند شبکه‌های هوشمند و حفاظت ازتاسیسات پشتیبانی می‌کند. عدم امنیت و حریم خصوصی در تصویب اینترنت اشیا توسط شرکت‌کننده‌ها و افراد، مقاومت ایجاد خواهد کرد.

اینترنت اشیا طی چند سال آینده با رشد بیش از ۳۰ درصد ادامه خواهد یافت. طبق پیش‌بینی محققان دپارتمان تحقیق Statista [6] تا سال ۲۰۲۵ تعداد کل دستگاه‌ها IoT به ۷۵ میلیارد دستگاه و یا به گفته IDC [7]، به ۸۰میلیارد دستگاه افزایش می‌یابد . تعداد دستگاه‌ها و موارد استفاده ازIoT همچنان رشد می‌کند، اما سازمان‌ها باید با چالش‌های امنیتی که از ابتدا چالش‌برانگیز بوده است را برطرف نمایند.

امنیت به‌عنوان یک از موانع در توسعه سیستم‌های مبتنی بر اینترنت اشیا در نظر گرفته می‌شود، همه می‌خواهند امنیت را در این سیستم‌ها افزایش دهند، اما هنگامی که هزینه آن را می‌فهمند، منصرف می‌شوند. اگرچه فروشندگان این سیستم‌ها در زمینه امنیت سرمایه‌گذاری می‌کنند، اما همیشه نمای کلی معماری یا توپولوژی سطح تهدید را برنامه‌ریزی نمی‌کنند و همه فروشندگان یا شرکا مهارت‌های امنیتی مناسبی ندارند. همچنین، سازمان‌ها اغلب بدون اعمال کامل تدابیر امنیتی لازم سیستم‌های خود را گسترش می‌دهند. آینده IoT همیشه با چالش امنیتی آن مطرح می‌شود [8].

احرازهویت و امنیت در مدارات الکترونیکی و سیستم‌های اینترنت اشیا جزو مهم‌ترین مباحث در این حوزه می‌باشد. شکل (۱-۲) که توسط نرم‌افزار VOSviewer با استفاده از واژه‌های کلیدی

مقالات رسم شده به خوبی اهمیت موضوع امنیت و احرازهویت را نشان می‌دهد، سایز هر نود نشـانه محبوبیت آن در مطالعات انجام شده اخیر می‌باشد.



شکل (۱-۲) حوزه‌های مطالعاتی محبوب در سالهای اخیر گزارش شده توسط [9]

بر اساس گزارش Statista [6]، پیش‌بینی می‌شود تا سال ۲۰۲۵ اینترنت اشیا بـا ۷۵ میلیـارد دستگاه در سراسر جهان به اینترنت متصل شود؛ اما چنین دستگاه‌هایی می‌تواننـد در صـورت عـدم امنیت با تهدیدات و مخاطراتی برای زیرساخت‌های ملی همراه باشند. درواقع، گزارش‌های مربوط بـه خرابی‌های امنیت اینترنت اشیاء فراوان است. دولت‌هـای سراسـر جهـان از خطـرات احتمـالی آگـاه هستند و برای کاهش ایـن مشـکلات اقـداماتی اتخـاذ می‌کننـد. بـه عنـوان مثـال، دولـت اسـترالیا پیش‌نویس یک قانون داوطلبانه برای امنیت در صنعت IoT [10]را در اوایـل سـال ۲۰۲۰ مـیلادی

معرفی کرد که شامل ۱۳ پیش‌نویس اصلی است، ازجمله اطمینان از یکپارچگی نرم‌افزار، اجرای سیاست افشای آسیب‌پذیری و به حداقل رساندن سطوح در معرض حمله که هدف آن حفاظت از داده‌ها در صنعت است.

۱-۴-۲- برای اطمینان از امنیت ارتباطات برای اینترنت اشیا چه باید کرد؟

اعتماد صفر

رویکرد سنتی برای امنیت شبکه این بود که دسترسی را به‌شدت کنترل کنید، اما به‌طور ضمنی فرض کنید که فضای داخلی شبکه یک مکان امن است. بارها و بارها، این یک فرض نادرست بوده است. در مقابل، رویکرد اعتماد صفر، اعتماد کور را با تأیید و ضمانت‌های رمزنگاری قوی جایگزین می‌کند. هدف این است که حتی در صورت به خطر افتادن زیرساخت‌های زیرین، امنیت، یکپارچگی و حریم خصوصی حفظ شود. اعتماد صفر هیچ وضعیت خاصی برای شبکه ندارد و با آن مانند اینترنت عمومی رفتار می‌کند.

محرمانه بودن

یکی از الزامات اساسی سیستم‌های ارتباطی، چه برای کاربردهای شخصی و چه صنعتی، این است که اشخاص ثالث نتوانند داده‌های ارسال شده از طریق شبکه را شناسایی کنند. همه داده‌ها باید از نقطه‌ای که تولید می‌شود به هر جایی که منتقل می‌شود رمزگذاری شود. رمزگذاری سرتاسری باید به‌طور قوی، دقیق و استاندارد با الگوریتم‌های منحصر به فرد انجام شود. این اصل به‌گونه‌ای تنظیم شده است که حتی اگر یک شنودگر به داده‌ها دسترسی داشته باشد، رمزگذاری سرتاسری محرمانه بودن داده‌ها را تضمین می‌کند. سیستم‌ها باید استفاده از رمزگذاری را برای کاربران آسان کرده و به صورت پیش‌فرض آن را فعال کنند.

تمامیت و درستی

علاوه بر اطمینان از محرمانه بودن، ضروری است که مهاجم نتواند پیام‌ها را با موفقیت دستکاری یا جعل کند. چنین تلاش‌هایی باید توسط شبکه قابل‌شناسایی باشد. همچنین نباید توانایی پخش مجدد یک پیام ضبط شده و احرازهویت آن با موفقیت وجود داشته باشد. رویکردهای رمزنگاری قوی، مانند احرازهویت پیام مبتنی بر PUF می‌تواند این محافظت‌ها را ارائه دهد و باید برای اطمینان از یکپارچگی داده‌ها مورد استفاده قرار گیرد.

حریم خصوصی

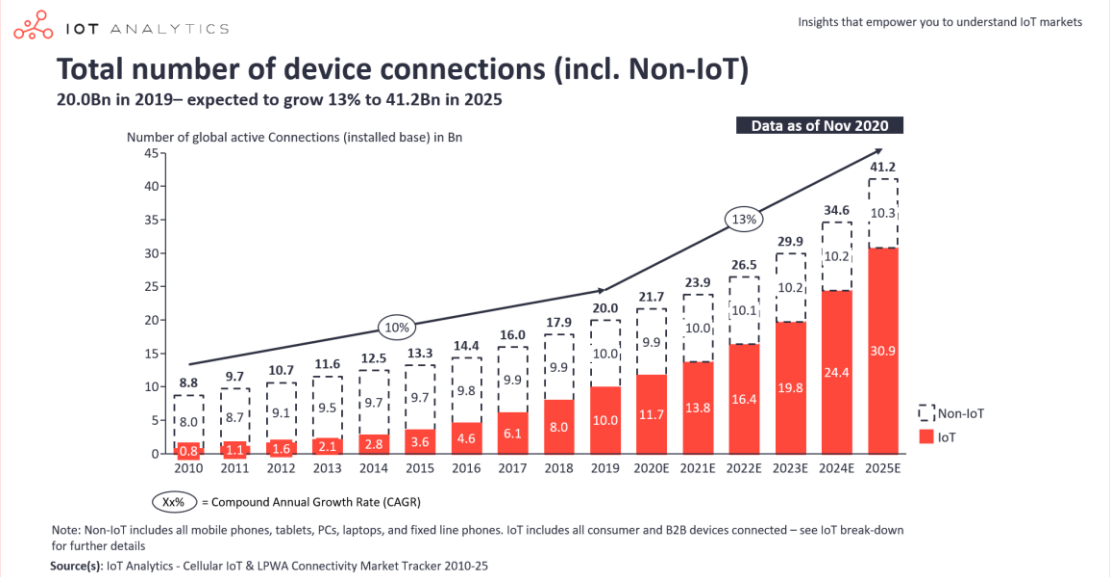
وقتی صحبت از حفظ حریم خصوصی می‌شود، این مهم است که به اشخاص ثالث این امکان داده نشود که هویت خود را بتوانند تعیین کنند، یا این توانایی را داشته باشند که بدانند پیام‌ها از یک دستگاه یا چند دستگاه ارسال می‌شوند. در حالی که استفاده از اصول رمزنگاری قوی برای رمزگذاری و یکپارچگی داده‌ها نسبتاً مناسب است، اما باید با استفاده از احرازهویت این امنیت را ارتقا داد. با اتخاذ این اقدامات احتیاطی، می‌توان تلاش هکرها برای حمله به یک دستگاه یا کاربر خاص را خنثی کرد.

مقیاس‌پذیری

اینترنت اشیا با قابلیت اتصال میلیاردها دستگاه، باید مقیاس‌پذیر باشد تا نه تنها موفقیت آینده خود را تضمین کند، بلکه امنیت آن را نیز تضمین کند. دستگاه‌های کم هزینه یا کم مصرف بهانه‌ای برای امنیت یا حریم خصوصی غیرموثر نیستند. صنعت اینترنت اشیا به سادگی باید به رشد نمایی بپردازد، در حالی که از کارکرد کم مصرف و کم هزینه نیز پشتیبانی می‌کند. اقدامات امنیتی پیش‌رو باید به کار گرفته شوند تا ارتقاء و پیشرفت‌ها حاصل شود و راه‌هایی برای رسیدگی به مسائل

امنیتی اجتناب‌ناپذیر ارائه شود. اگر صنعت، امنیت اینترنت اشیا را به درستی به کار نگیرد، امکان کنترل هکرها بر اطلاعات شخصی، تجاری یا ملی فوق‌العاده حساس می‌تواند پیامدهای مخربی داشته باشد [11].

استفاده از PUF در احرازهویت پروتکل‌های ارتباطی می‌تواند باعث بازگشت امنیت در مخابره و جلوگیری از استفاده غیرقانونی مدارات مجتمع شود. پیاده‌سازی سخت‌افزاری این معماری خود با چالش‌هایی از جمله انرژی و فضای پیاده‌سازی که بسیار مهم و تأثیرگذار هستند، روبرو است. از جمله اهداف این پایان‌نامه بررسی روش‌های مختلف حمله و پیاده‌سازی حملات مبتنی بر یادگیری ماشین و ارائه معماری نوین با دید سخت‌افزاری است که منجر به بهبود مقابله با این‌گونه حملات در بحث احرازهویت در مدارات مجتمع خواهد شد.



شکل (۱-۳) تخمین آینده تعداد دستگاه‌های متصل به اینترنت‌اشیا توسط مرجع [12]

۱-۵- خلاصه فصل‌ها

در فصل ۲ به بیان مفاهیم پیش‌نیاز در این حوزه می‌پردازیم. در ابتدا بـه توصـیف، کـاربرد و بررسی انواع PUF خواهیم پرداخت. سپس به بررسی انواع حملات در حوزه اینترنت‌اشـیا و معرفـی آن‌ها می‌پردازیم. در نهایت به بررسی انواع روش‌های یادگیری ماشین مورد استفاده می‌پردازیم.

در فصل ۳ مروری بر بررسی تحقیقات صورت گرفته قبلی خواهیم داشت، این‌رو ضمن معرفی معماری‌های مختلف کارهایی را بررسی می‌کنیم که امنیت این مدل‌ها را زیر سؤال برده‌اند.

در فصل ۴ راهکارهای پیشنهادی برای حمله به دو مدل از قوی‌ترین PUF-ها ارائه می‌دهیم، سپس با بررسی آن‌ها و بررسی عمیق‌تر مدل اصلی اقدام به ارائه یک مدل جدیـد بـا امنیـت بیشـتر می‌کنیم.

در فصل ۵ به بررسی نتـایج بـه دسـت آمـده از راهکارهـای حملـه بـه دو مـدل قـوی PUF می‌پردازیم، سپس مدل خود را کاملاً بررسی می‌کنیم و نتایج را گزارش می‌کنیم.

فصل ۲: مفاهیم پیش‌نیاز

۲-۱- مقدمه

در این فصل به بررسی مفاهیم پیش‌نیاز خواهیم پرداخت. در ابتدا توضیحات کلی در خصوص PUF و مدل‌های آن ارائه خواهیم نمود، سپس در ادامه انواع حملات در حوزه امنیت و احرازهویت را بررسی خواهیم نمود و انواع روش‌های حمله یادگیری ماشین موفق را شرح خواهیم داد.

۲-۲- توابع غیرهمسان فیزیکی (PUF)

تاکنون تعاریف زیادی برای PUF ارائه شده است، در یکی از بهترین آن‌ها PUF را "معماری وابسته به ذات و غیرقابل نمونه‌برداری و خاص برای هر نمونه با استفاده از خواص ذاتی اشیا" تعریف می‌کند. بدین معنی که یک معماری یکسان که به ویژگی‌های فیزیکی و ذاتی مدار مجتمع وابسته است و با توجه به اینکه این ویژگی‌های حین ساخت به‌صورت تصادفی بوده، پس برای هر نمونه، منحصر به فرد است و با توجه به اینکه خارج از کنترل ما هستند، پس قابل نمونه‌برداری نیست. این مفهوم را می‌توان مانند خاصیت بیولوژیکی اثرانگشت برای انسان تشبیه نمود.

دلایل این موضوع به شرح زیر است:

• اثرانگشت یکی از ویژگی‌های ذاتی هر انسان است، بدین معنا که هر انسان با اثرانگشت خاص خود به دنیا می‌آید. ویژگی‌های ذاتی با سایر ویژگی‌های هویتی همچون اسم، کدملی، امضا تفاوت دارد؛ زیرا این موارد پس از تولد انسان تعیین می‌شوند و قابلیت تغییر دادن را دارند، اما اثرانگشت منحصر به هر انسان بوده و قابل‌تغییر نیست.

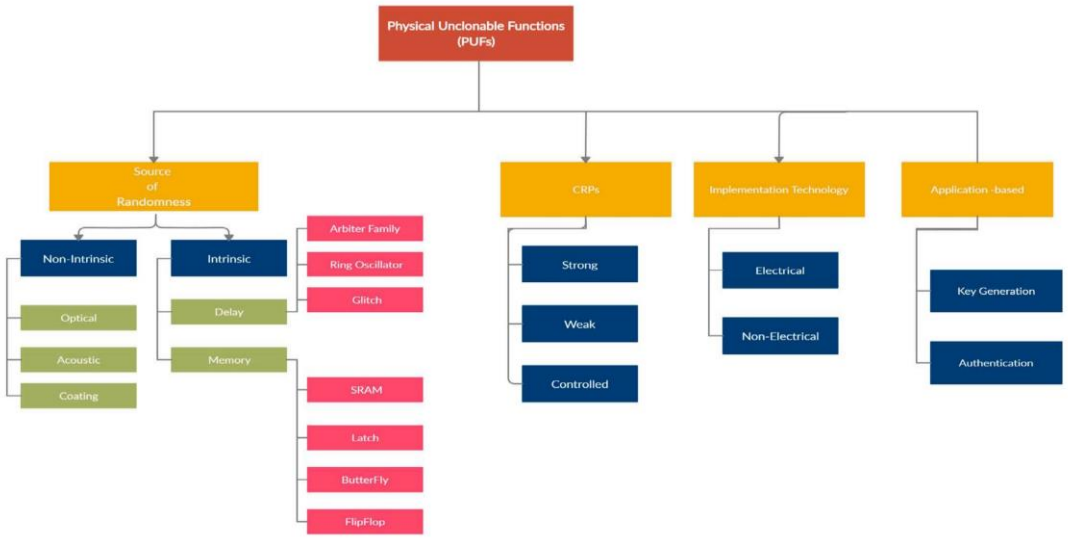
• اثرانگشت برای انسان‌ها یک ویژگی فردی است، نه ویژگی حیاتی؛ به عبارت دیگر هر انسان باید دو پا، دو دست، دو چشم و... را داشته باشد. ولی اثرانگشت در قبال پرسش "انسان بودن یا نبودن؟" مطرح نمی‌شود، بلکه در برابر سؤال "کدام یک از انسان‌ها؟" مطرح می‌شود و مشخص کننده نمونه منحصر به فرد از مجموعه است.

• اثرانگشت برای تمامی انسان‌ها با هر انسان دیگری متفاوت می‌باشد، به عبارتی دیگر هر انسان یک اثرانگشت خاص خود را دارد و هیچ دو فردی یک اثرانگشت مشابه را ندارند. PUF نیز برای هر مدار یک پاسخ داشته و برای هرکدام متفاوت است، پس از این لحاظ نیز مشابه اثرانگشت است.

مفهوم PUF در اوایل قرن بیست و یکم میلادی به این صورت مطرح شد، اما کارهایی در این حوزه قبل از این مفهوم انجام‌شده بود و فقط به این صورت ذکر نشده بودند، همچنین بعد از این نام‌گذاری نیز برخی از گروه‌ها کارهایی در این زمینه انجام داده‌اند، اما به دلیل عدم آشنایی با این حوزه در این زمینه ذکر نشده‌اند. کارهای متنوع و فراوانی در این حوزه در حال انجام است در ادامه به بررسی مهم‌ترین موارد در این حوزه می‌پردازیم[13].

۲-۲-۱- تقسیم بندی PUFها

شکل (۲-۱) انواع دسته‌بندی PUF-ها را مطرح کرده است. در این بخش برخی از تقسیم‌بندی‌های انواع PUF-ها را بررسی می‌کنیم:



شکل (۲-۱) انواع دسته‌بندی برای PUFها [14]

الکترونیک

با توجه به اینکه PUF-ها از مواد مختلفی مانند پلاستیک، کاغذ قطعات الکترونیک و مدار مجتمع‌های سیلیکونی ساخته می‌شوند، یا به عبارت با توجه به تکنولوژی پیاده‌سازی آن‌ها، می‌توان آن را به دودسته زیر تقسیم نمود:

غیر الکترونیک(Non-electronic PUF)

PUF-هایی که بر پایه ذات ماده از خود این خاصیت تصادفی بودن را نشان می‌دهند؛ مثلاً خاصیت پخشی مواد یا بازتابش تصادفی نور از کاغذ و ... .

الکترونیک(Electronic PUF)

PUF-هایی که این تغییرات تصادفی بر پایه خاصیت الکترونیکی آن‌ها وجود دارد. برای مثال خازن، مقاومت و... که معمولاً این تغییرات با استفاده از ابزار الکترونیکی سنجیده می‌شود. البته بعضی موارد هستند که به‌صراحت نمی‌توان آن را الکترونیکی و یا غیر الکترونیکی نامید.PUF-های سیلیکونی[1] را که اغلب PUF-های الکترونیکی را تشکیل می‌دهند را نیز می‌توان جزو این دسته برشمرد، مثلاً مدار مجتمع الکترونیک که خاصیت PUF دارد و در مدار مجتمع سیلیکونی کشت می‌شود.

ذاتی یا غیر ذاتی

PUF-ها از نظر خاصیت تصادفی بودن و منبع به ذاتی یا غیر ذاتی تقسیم می‌شوند، درصورتی که دو شرط زیر را داشته باشد به آن PUF ذاتی می‌گوییم:

۱-ارزیابی آن به‌صورت داخلی اندازه‌گیری شود.

۲-خاصیت تصادفی مخصوص به هر نمونه در فرآیند ساخت آن مشخص شود.

قوی یا ضعیف یا کنترل شده

به یک PUF زمانی قوی[2] می‌گوییم که حتی اگر برای بازه زمانی PUF در اختیار مهاجم باشد باز چالش-پاسخ‌هایی هنوز وجود داشته باشد که مهاجم به آن دسترسی نداشته باشد. اگر تعداد CRP-های یک PUF خیلی زیاد باشد و پیاده‌سازی مدل PUF دقیق بر پایه CRP-ها ممکن نباشد، می‌توان آن PUF را قوی نامید.

دسته دیگری که به تازگی برای PUF-ها مطرح‌شده کنترل‌شده می‌باشد، این دسته به عبارتی

[1] Silicon

[2] Strong

زیرمجموعه PUF-های قوی نیز می‌باشد، زمانی که برای مخفی کردن چالش-پاسخ‌های یک PUF قوی از یک مدار کنترل‌کننده استفاده کنند می‌توان آن‌ها را در دسته PUF-های کنترل‌شده جای داد [13], [14].

۲-۲-۲- انواع PUF

در این بخش به بررسی انواع معماری‌های ارائه‌شده PUF می‌پردازیم.

Arbiter PUF

اولین نوع PUF ارائه‌شده مبتنی بر تأخیر مسیر[1] Arbiter PUF می‌باشد. ایده طراحی این نوع PUF وضعیت رقابتی[2] بین دو مسیر دیجیتال را در مدار مجتمع سیلیکونی مطرح می‌کند. کارکرد A-PUF بدین صورت است که مقدار قرار گرفته در ورودی از طریق دو مسیر مختلف با تأخیر متفاوت به Arbiter خواهند رسید که Arbiter بر اساس ویژگی‌های از قبل تنظیم‌شده یکی را انتخاب خواهد کرد.

با توجه به این مورد که تأخیر در مسیرهای دیجیتال متأثر از تغییرات فرآیند ساخت[3] هستند، بنابراین تأخیرها را می‌توان تصادفی و منحصر به هر دستگاه دانست.

در حالت خاص در A-PUF ممکن است هر دو مسیر به‌صورت هم‌زمان به Arbiter برسند، به‌عبارت‌دیگر تأخیر برابر داشته باشند، بنابراین Arbiter وارد حالت ناپایدار[4] شده و بعد از مدت کوتاهی جوابی تصادفی تولید می‌کند. به این پدیده غیرقابل اطمینان بودن[5] گفته می‌شود.

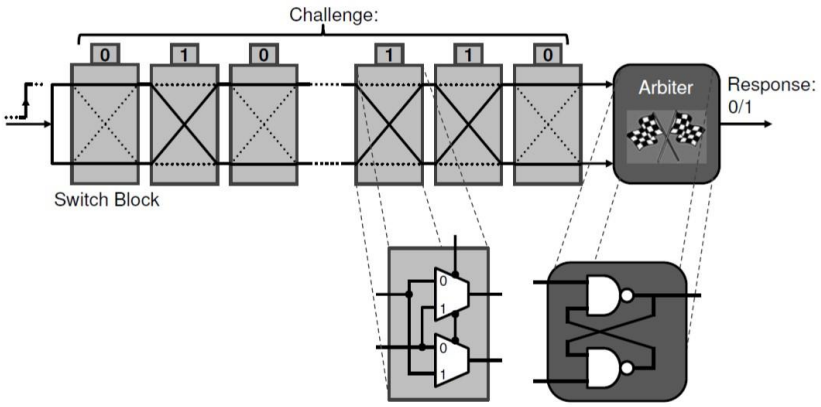
[1] Delay Path
[2] Race Condition
[3] Process Variation
[4] Metastable
[5] Unreliability

زمانی می‌توانیم جواب Arbiter را کاملاً تصادفی بدانیم که:

۱-تأخیر خطوط متقارن و متفاوت باشند، به عبارت دیگر هر تأخیر متفاوت به علت فرآیند ساخت ایجادشده باشد.

۲-مدار Arbiter کاملاً دقیق عمل کند و عادل باشد.

نمونه APUF پیاده‌سازی شده در شکل (۲-۲) مشاهده می‌شود که بر اساس یک سویچ رقابت آغاز می‌شود و بر اساس مقدار چالش‌های قرارگرفته روی Select در MUX-ها ممکن است هرکدام از مسیرها زودتر به Arbiter برسند.



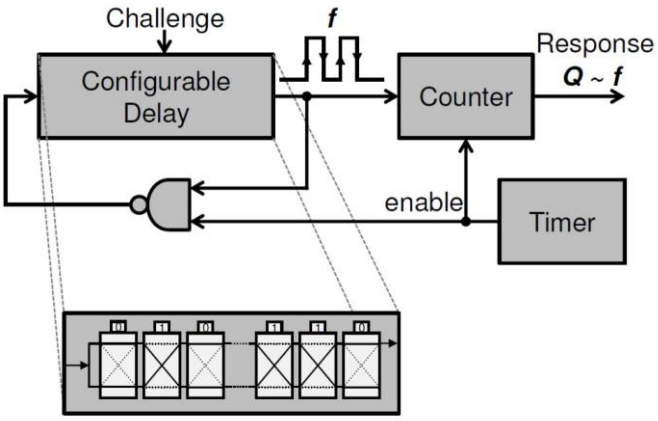
شکل (۲-۲) شمای کلی APUF [13]

با استفاده از الگوریتم‌های یادگیری ماشین به‌ویژه الگوریتم‌های SVM[1], LR[2], ANN[3] حملات مختلفی بر روی PUF-ها صورت گرفته است. کارهای مختلفی نیز جهت بالابردن امنیت و مقابله با حملات مدل‌سازی در APUF انجام شده است که می‌توان به غیرخطی کردن مسیر تأخیر با اضافه کردن سیگنال‌های کنترلی در ورودی و خروجی PUF اشاره کرد، در ادامه به بررسی بیشتر این نوع PUFها و حملات آن‌ها خواهیم پرداخت.

[1] Support Vector Machine
[2] Logistic Regression
[3] Artifitial Neural Network

Ring Oscillator PUF

نوع بعدی، PUFهای بر پایه مسیر تأخیر، ROPUF نامیده می‌شود. به‌صورت عادی ROPUF در طرح اولیه شامل دو قسمت شمارنده فرکانس[1] و مدار نوسان‌ساز[2] می‌باشد که برای محاسبه خروجی از روشی مشابه APUF استفاده می‌کنند همچنین به دلیل اثرگذاری بیش از حد عوامل محیطی همچون دما و نویز و ولتاژ برای ثابت ماندن مقدار در هر دستگاه از دو شمارنده استفاده می‌شود که نسبت خروجی این دو شمارنده را برهم تقسیم کرده و بر این اساس خروجی متناسب را تولید می‌کند.

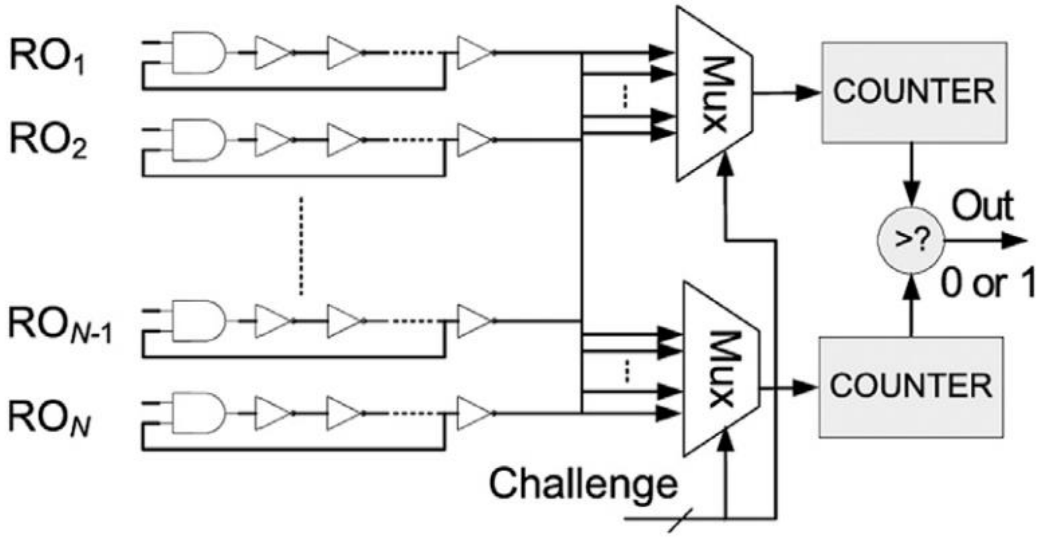


شکل (۲-۳) طرح اولیه ROPUF [13]

پروتکل ROPUF مطرح شده در برابر حملات مدل‌سازی مقاوم نبود، پس با ایجاد تغییر در ساختار اصلی طرح شکل (۲-۴) برای مقاوم‌سازی ROPUF ها ارائه شد، این طرح نیز همانند ROPUF اولیه عمل می‌کند، با این تفاوت که تعداد زیادی نوسانگر با گیت‌های معکوس کننده پیاده‌سازی شده است که بر اساس چالش[3] یکی از آن‌ها انتخاب می‌شود و به شمارنده اعمال می‌شود. برای پیاده‌سازی این مدل می‌توان حالت‌های مختلف را در نظر گرفت. برای تولید پاسخ[4] نیز می‌توان تمامی حالات n را در نظر گرفت که این عمل باعث بیشتر شدن تعداد جفت پاسخ‌ها خواهد شد اما

[1] Frequency Counter
[2] Ring Oscillator
[3] Challenge
[4] Response

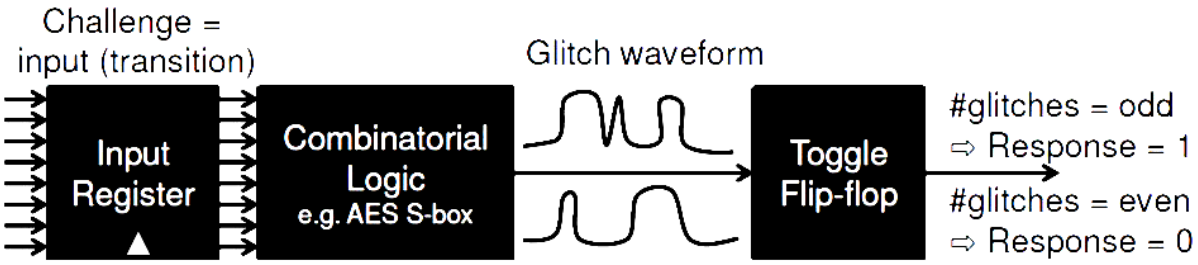
وابستگی بین پاسخ‌ها را افزایش می‌دهد که بهتر است برای کمتر شدن هم‌پوشانی به دو قسمت تقسیم کرده و مستقل از هم در نظر بگیریم[15].



شکل (۲-۴) شمای کلی طرح دوم ارائه شده ROPUF [16]

Glitch PUF

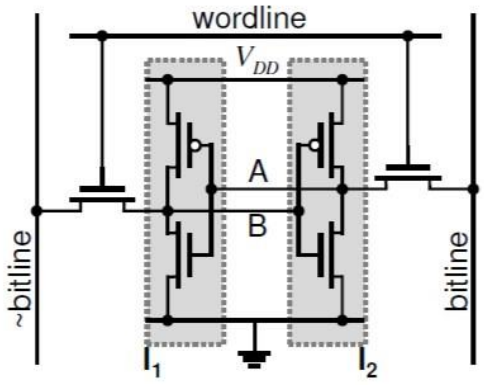
نوع دیگری از PUF مبتنی بر تأخیر مسیر را Glitch PUF می‌نامند. رفتار این نوع PUF برپایه پدیده GLITCH در مدارات ترکیبی دیجیتال می‌باشد. در مدارات ترکیبی دیجیتال هنگام اعمال ورودی‌ها با توجه به تأخیرهای مسیرها مدار ابتدا وارد حالت ناپایدار می‌شود و پس از مدتی مقدار واقعی خود را می‌گیرد. این اتفاق را در مدارات GLITCH می‌نامند. با توجه به اینکه تأخیر مسیرها بر اساس تغییرات فرآیند ساخت متفاوت، تصادفی و منحصر به هر دستگاه است، پس می‌توان از آن جهت کاربرد به عنوان پاسخ PUF استفاده کرد. درشکل (۲-۵) شمای کلی PARITY GLITCH PUF نمونه‌ای از GLITCH PUF آورده شده است که بر اساس تعداد glitchهای اتفاق افتاده پاسخ موردنظر را تولید می‌کند[13].



شکل (۲-۵) شمای کلی PARITY GLITCH PUF [13]

SRAM PUF

نوع دیگری از PUF ها را SRAM-PUF می‌نامند که بر پایه بلوک‌های حافظه[1] عمـل می‌کنـد، در شکل (۲-۶) یک بلوک حافظه SRAM بر پایه مدار bistable با تکنولوژی cmos را نشان داده شده است. همان‌طور که پیداست مدار از شش ترانزیستور تشکیل شده است، با بهره از بـازخورد مثبـت در مدار، مقدار حافظه تقویت می‌شود (شکل (۲-۷) نمونه‌ای از این تقویت را نمایش می‌دهـد). SRAM حافظه‌ای از نوع Volatile می‌باشد، پس با قطع برق مقدار آن پاک خواهـد شـد. مقـدار آن به‌صـورت تصادفی و بر اساس اثر گیت‌های معکوس کننده به وجود می‌آید (شکل (۲-۸))[17].



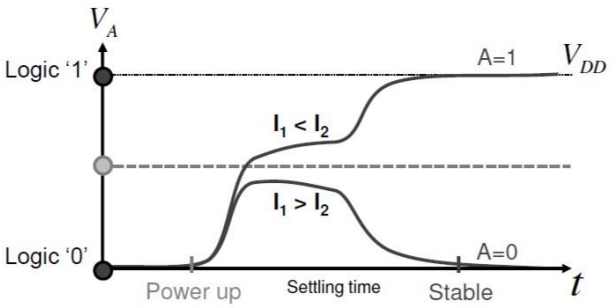
شکل (۲-۶) یک بلوک سلول حافظهSRAM [13]

مدار SRAM سه حالت مختلف دارد که دو حالت پایدار و یکی ناپایدار دارد ولی مدار هرگز در

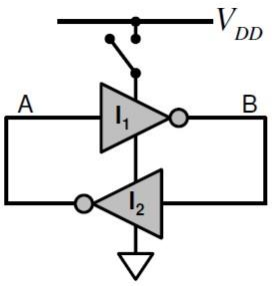
[1] Memory

حالت ناپایدار نخواهد ماند، در صورتی که مدار به هر دلیلی مثل وجود نویز در مدار یـا روشـن شـدن مدار به حالت ناپایدار درآید، بنا به‌سرعت مدار به‌صورت تصادفی به یکی از حالت‌های پایدار می‌رود.

اصول کار SRAM PUF مبتنی بر رفتار حالت گذرا در بلوک SRAM در زمـان روشـن شـدن مدار می‌باشد. زمانی که مدار روشن می‌شود یا بـه عبـارتی Vdd مـدار متصـل می‌شـود، طبـق اینکـه کدام‌یک از دو گیت معکوس کننده پرقدرت‌تر هستند مدار به یکی از دو حالت مـی‌رود. هرچقـدر کـه این دو گیت معکوس کننده مشابه یکدیگر ساخته شوند، تأثیر تغییرات فرآیند ساخت بر ایـن انتخـاب بیشتر می‌شود.



شکل (۲-۷) نمودار اثر قدرت گیت ها در مقدار اولیه سلول SRAM [13]

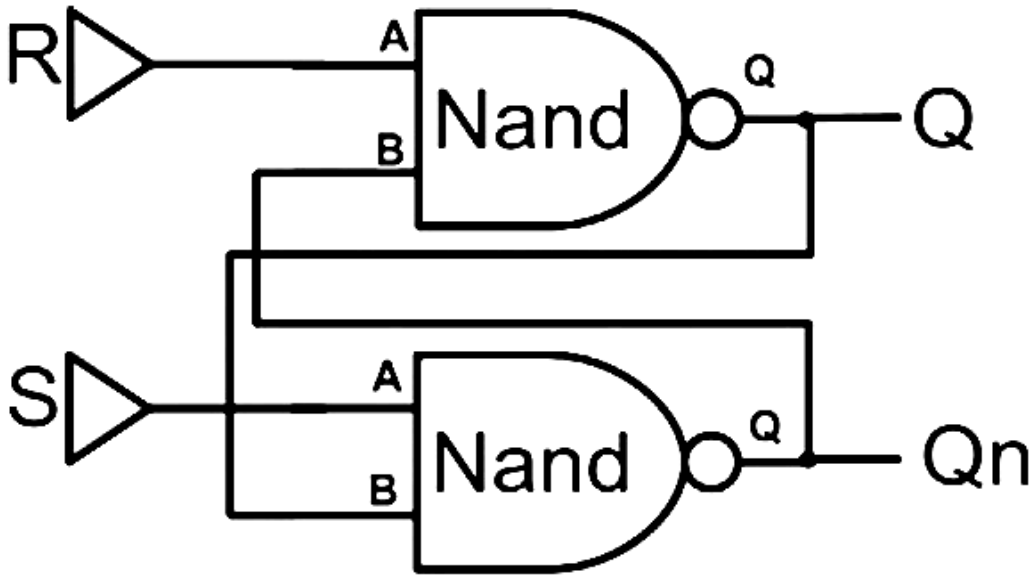


شکل (۲-۸) بلوک دیاگرام سلول SRAM [13]

این تغییرات برای هر سلول به شکل تصادفی می‌باشـد و بـا توجـه بـه تعـداد زیـاد سـلول‌های SRAM می‌توان این نوع PUF را از نوع PUF-های قوی دانست. علاوه بر SRAM مدل‌های گوناگونی از PUF های مبتنی برbistable وجود دارند که در ادامه به بررسی برخی از آن‌ها می‌پردازیم:

LATCH PUFs

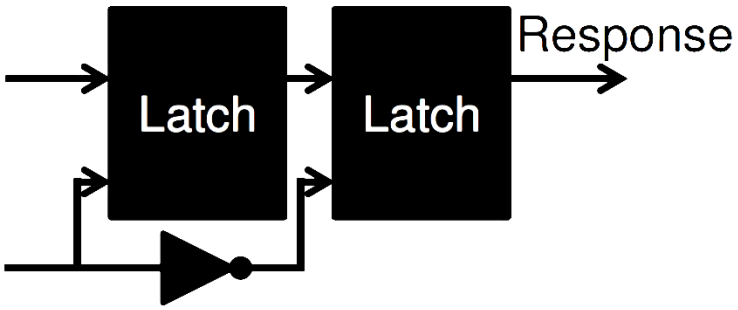
تقریباً فرآیندی مشابه SRAM دارد با این تفاوت که محدود به زمان روشن شدن مدار (اتصال ولتاژ) نمی‌باشد و در هر زمانی می‌توان از آن استفاده نمود که کاربردهای خاص خود را دارد[18].



شکل (۲-۹) شمای کلی LATCH PUF [19]

FLIP-FLOP PUFs

در شکل (۲-۱۰) نوعی ساختار PUF بر پایه bistable مبتنی بر اتصال دو لـچ[1] ارائـه گردیـده است. عملکرد آن بر اساس سیگنال اعمال شده به آن تعیین می‌شود[20].

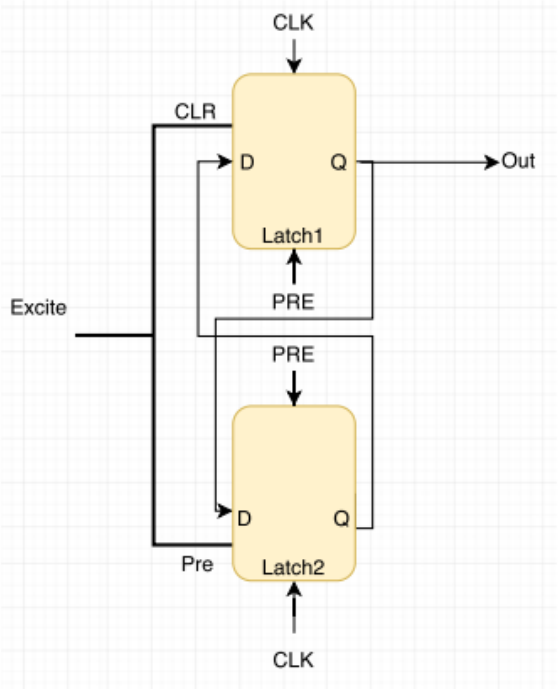


شکل (۲-۱۰) شمای کلی FLIP-FLOP PUF [13]

[1] Latch

BUTTERFLY PUFs

با توجه به اینکه اکثر FPGA-ها بعد از روشن شدن مدار مجتمع بلافاصله تمامی خانه‌های SRAM را پاک می‌شوند، پس پیاده‌سازی SRAM PUF در FPGA-ها تقریباً غیرممکن است؛ بنابراین ساختاری استفاده از لچها پیاده‌سازی شده است که می‌تواند تمامی قابلیت‌های SRAM را داشته باشد و همچنین مختص زمان روشن شدن نباشد[13].

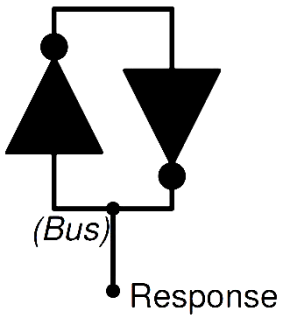


شکل (۲-۱۱) شمای کلی butterfly PUF [4]

BUSKEEPER PUFs

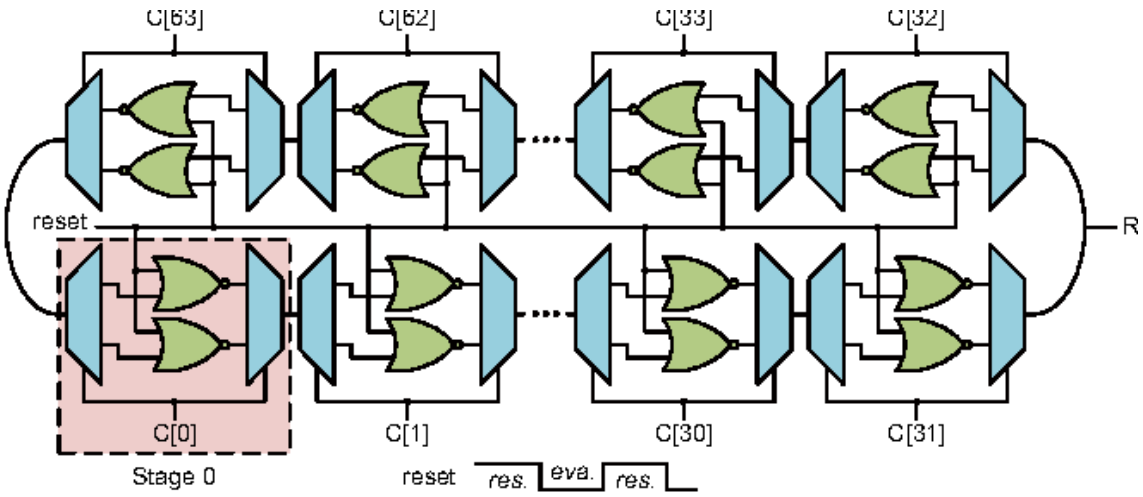
این نوع PUF-ها مبتنی بر bistable می‌باشند که با استفاده از مقدار bus مدار کنترل می‌شود. مزیت‌های استفاده از این نوع PUF را می‌توان به سادگی مدار و فضای اشغالی کمتر[1] آن دانست[13] .

[1] Area Efficient



شکل (۲-۱۲) شمای کلی BUSKEEPER PUF [13]

این نوع PUF مبتنی بـر SRAM PUF & RING OSCILLATOR PUF هـا هسـتند. بـدین ترتیب که شمای کلی همان شمای RING OSCILLATOR ها می‌باشد با این تفاوت کـه گیت‌هـای معکوس کننده بجای استفاده برای تولید حلقه نوسان، برای کاربرد bistable تعبیه شده‌اند[21].



شکل (۲-۱۳) شمای کلی BISTABLE RING PUF [22]

۲-۳- انواع حملات به شبکه‌های اینترنت‌اشیا در حوزه احراز هویت

به هرگونه حمله که هدف آن سرقت اطلاعات، تخریب و ... از شبکه محلی، اینترنت یا کامپیوتر

شخصی که از طریق مختلف صورت می‌گیرد حملات سایبری[1] گویند. حملاتMan-in-the-Middle، Password،Eavesdropping حملات برای جعل احرازهویت هستند که در ادامه به آن‌ها می‌پردازیم.

۲-۳-۱- حمله MitM

حمله Man-in-the-Middle یا MitM بدین صورت است که حمله کننـده خـود را در ارتبـاط مابین فرستنده و گیرنده وارد می‌کند. از انواع معمول این حملات می‌توان به این موارد زیر اشاره نمود:

Session hijacking

در این نوع حملات، حمله کننده فاصله‌ای بین فرستنده و گیرنـده ایجـاد می‌کنـد و بـا ربـودن آدرس IP فرستنده ارتباط را با گیرنده برقرار می‌نماید. سپس حمله کننده اطلاعات را از گیرنده گرفته و با ایجاد تغییرات لازم آن‌ها را به سرور مرکزی ارسال می‌کند. همچنین می‌تواند با ارسال دسـتوراتی باعث ایجاد تداخل در فرستنده گردد.

IP Spoofing

در این نوع حملات، حمله کننده با استفاده از آدرس IP فرستنده‌ای که توسط سرور مورد تأیید بوده و احرازهویت گردیده است، داده‌هایی را به سرور ارسال می‌کند کـه می‌توانـد آثـار مخربـی را بـا اجرای آن دستورات به سرور وارد نماید.

Replay

در این نوع حملات پس از ذخیره پیام‌های گذشته بین فرستنده و گیرنده بود و تکرار آن‌هـا در زمان موردنظر حمله‌کننده منجر به جعل هویت یکی از گره‌ها می‌گردد. این نوع حملات می‌تواند اثرات

[1] Cyber Attack

مخرب زیادی به دنبال داشته باشد و به‌راحتی رخ می‌دهد.

۲-۳-۲- Password attack

در بسیاری از شبکه‌ها، یکی از اساسی‌ترین مکانیزه‌های احرازهویت، استفاده از رمز عبور است. در صورت یافتن رمز عبور فرد یا دستگاه می‌تواند عملیات مخرب خود برای گره مربوطه را سرور انجام دهد. روش‌های زیادی جهت ایمن کردن رمز عبور در دو طرف ارائه گردیده‌اند اما همچنان در مخابره اطلاعات ضعف شدیدی وجود داشته و احتمال شنود و برداشت وجود دارد[23].

۲-۳-۳- Eavesdropping

در این نوع حمله، حمله‌کننده به‌وسیله شنود ارتباط و جمع‌آوری اطاعات مخابره شده، می‌تواند اطلاعاتی مثل شماره کارت، رمز عبور و اطلاعات محرمانه فرستنده را دریافت می‌کند و از آن‌ها سوءاستفاده می‌نماید. دو نوع حمله شنود وجود دارد که شامل فعال[1] و غیرفعال[2] می‌باشد بدین مفهوم که:

حمله غیرفعال: حمله‌کننده با شنود ارتباط، اطلاعات را دریافت می‌نماید.

حمله فعال: حمله‌کننده اطلاعات را به‌صورت آنلاین توسط معرفی خود به‌عنوان گره دوست که به‌وسیله ارسال درخواست به سرور صورت می‌گیرد جمع‌آوری می‌کند. این کار به پروب کردن و یا حمله تهاجمی[3] نیز معروف است[24].

[1] Active
[2] Passive
[3] Tamper Attack

۲-۴- الگوریتم‌های یادگیری ماشین

در این بخش الگوریتم‌های یادگیری ماشین که در ادامه از آن‌ها استفاده کرده‌ایم را به‌طور خلاصه معرفی می‌کنیم.

۲-۴-۱- رگرسیون لجستیک[1]

یکی از روش‌های دسته‌بندی[2] در مبحث آموزش نظارت‌شده[3] رگرسیون لجستیک است. در این روش رگرسیونی، از مفهوم و شیوه محاسبه نسبت بخت[4] استفاده می‌شود.

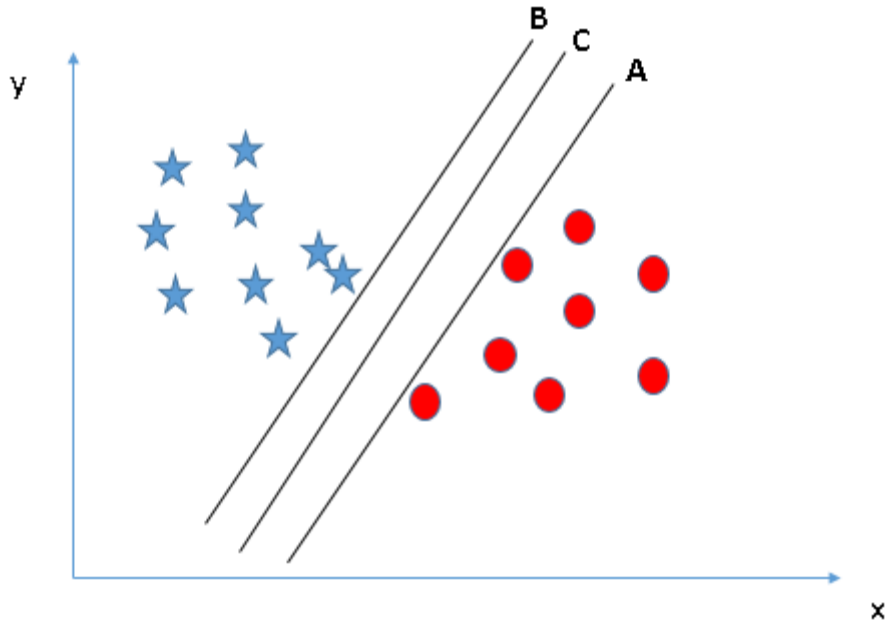
این الگوریتم مربوط به کلاس‌بندی است و برای برآورد مقادیر گسسته (مقادیر باینری مانند ۰/۱، آری/خیر، درست/نادرست) بر اساس مجموعه‌ای از متغیرهای وابسته استفاده می‌شود. به بیان ساده‌تر این الگوریتم احتمال رخداد یک رویداد را برحسب گنجاندن داده‌ها در یک تابع لجستیکی پیش‌بینی می‌کند. ازاین‌رو به نام رگرسیون logit نیز شناخته می‌شود. از آنجا که این الگوریتم، احتمال را پیش‌بینی می‌کند، مقادیر خروجی آن بین ۰ تا ۱ هستند.

۲-۴-۲- ماشین بردار پشتیبان[5]

یک الگوریتم نظارت‌شده یادگیری ماشین است که هم برای مسائل طبقه‌بندی و هم مسائل رگرسیون قابل‌استفاده است؛ با این حال از آن بیشتر در مسائل طبقه‌بندی استفاده می‌شود. در

[1] Logistic Regression
[2] Classification
[3] Supervised Learning
[4] Odds Ratio
[5] Support Vector Machine

الگوریتم SVM، هر نمونه داده را به عنوان یک نقطه در فضای n بعدی روی نمودار پراکندگی داده‌ها ترسیم کرده (n تعداد ویژگی‌هایی است که یک نمونه داده دارد) و مقدار هر ویژگی مربوط به داده‌ها، یکی از مؤلفه‌های مختصات نقطه روی نمودار را مشخص می‌کند. سپس، با ترسیم یک خط راست، داده‌های مختلف و متمایز از یکدیگر را دسته‌بندی می‌کند، نمونه این دسته‌بندی در شکل (۲-۱۴) آورده شده است.



شکل (۲-۱۴) نمونه دسته‌بندی ماشین بردار پشتیبان

۲-۴-۳- شبکه‌های عصبی عمیق[1]

یادگیری عمیق[2] بخشی از یادگیری ماشین[3] است که خود شاخه‌ای از هوش مصنوعی می‌باشد و راه‌حلی برای مسائل دسته‌بندی و حل مسائل غیرخطی ارائه می‌کند.

[1] Deep Neural Network (DNN)
[2] Deep Learning
[3] Machine Learning

شبکه‌های عصبی کانولوشن[1] نوع خاصی از شبکه‌های عصبی می‌باشد که بر مبنای یادگیری عمیق بوده و در سال‌های اخیر پیشرفت چشمگیری درزمینه یادگیری ماشین داشته است. در این شبکه‌ها به‌جای ضرب ماتریسی ساده از عملیات کانولوشن استفاده می‌شود که آن را برای بسیاری از کاربردهای عملی ایدئال می‌کند. استفاده از عملگر کانولوشن به‌شدت حجم محاسبات را کاهش می‌دهد و به ما اجازه می‌دهد، شبکه‌هایی با تعداد لایه‌های بالا داشته باشیم و درعین‌حال در یک‌زمان منطقی بتوانیم شبکه را آموزش دهیم و از آن خروجی بگیریم. بیشترین کاربرد این شبکه‌ها در سال‌های اخیر مربوط به پردازش تصویر، بینایی ماشین[2] و دسته‌بندی بوده است، به‌گونه‌ای که این حوزه‌ها با ورود شبکه‌های کانولوشن دگرگون‌شده است.

[1] Convolutional Neural Network (CNN)
[2] Computer Vision

فصل ۳: مروری بر پیشینه تحقیق

۳-۱- مقدمه

هدف در این فصل، ارائه مختصری از کارهای پیشین انجام‌شده در این زمینه و بررسی دست‌آوردهای هریک می‌باشد. همچنین دلایل و ضرورت انجام این تحقیق با مقایسه کارهای صورت گرفته، در بخش نتیجه‌گیری آورده شده است. در این بخش ابتدا به بررسی مراجع و منابع مرتبط با PUF می‌پردازیم. انواع پیاده‌سازی PUF، دسته‌بندی و بررسی کاربرد و میزان امنیت هرکدام به‌صورت کامل در کتاب [25] مورد تحلیل و بررسی قرارگرفته است. از این مرجع به‌عنوان مرجع پایه PUF استفاده خواهیم کرد.

امروزه مقوله امنیت در سیستم‌های الکترونیکی و رایانشی یکی از مهم‌ترین و حیاتی‌ترین موضوعات موردتحقیق است. تاکنون برای بالا بردن سطح امنیت مدارات و سیستم‌های الکترونیکی روش‌های مختلفی ارائه‌شده است. یکی از این روش‌ها که در حوزه امنیت سخت‌افزار بسیار پرکاربرد است و عمدتاً برای احرازهویت در وسایل الکترونیکی استفاده می‌شود، مفهوم PUF است. این مفهوم برای اولین بار در [16] مطرح شد اما دیری نگذشت که امنیت این ساختار توسط [26] موردحمله قرار گرفت، ازین پس انواع مختلفی از PUF ها برای جلوگیری از حملات ارائه شد، همچنین انواع مختلفی از مقالات به بررسی مقاومت انواع مدل‌های PUF پرداخته‌اند.

مقاله [4] برای اولین بار مفهوم PUF را در مدارات مجتمع سیلیکونی مطرح نمود. در این معماری دیگر کلید رمزنگاری در حافظه دیجیتال ذخیره نمی‌شد، پس می‌توان این معماری را در برابر حملات مستقیم و غیرمستقیم، در زمان خاموشی سیستم مقاوم دانست. بنا به کاربرد، پیاده‌سازی ASIC و یا FPGA و همچنین بهبود معماری‌ها ازلحاظ مساحت، توان مصرفی و رفع مشکلات موجود در معماری‌های گذشته، همواره ارائه معماری نوین PUF مدنظر محققان در این حوزه بوده است. از انواع مهم‌ترین و پرکاربردترین PUF ها می‌توان به جمله اسیلاتورهای زنجیره‌ای اشاره نمود. این معماری با توجه به نتایج مناسب، کاربردهای فراوانی هم در طراحی‌های ASIC و هم FPGA دارد[13]. از سایر ساختارها هم می‌توان به SRAM-PUF اشاره نمود که بر اساس رفتار تصادفی سلول‌های حافظه سیستم در زمان روشن شدن مدار عمل می‌نماید [27]. با توجه به اینکه در اکثر FPGAها خانه‌های حافظه بعد از روشن شدن مدار فوراً مقدار صفر به خود می‌گیرند، بنابراین استفاده از SRAM-PUF در این نوع FPGA ها با محدودیت‌هایی مواجه است، به همین دلیل، ساختاری بر همین پایه باکمی تغییرات ارائه‌شده است که مشکلات مدل قبلی را ندارد و محدود به زمان روشن شدن نیست، همچنین در هر زمان می‌توان از آن استفاده نمود. این ساختار با عنوان Butterfly-PUF پیشنهاد گردیده است[28].

با توجه به این ساختارها می‌توان PUFها را برحسب ویژگی‌های مشخصی، دسته‌بندی نمود. از مهم‌ترین این دسته‌بندی‌ها می‌توان به تعداد جفت چالش-پاسخ‌ها اشاره نمود. اگر فقط یک جفت چالش و پاسخ داشته باشیم در این صورت به این معماری PUF ضعیف گفته می‌شود ولی اگر تعداد جفت چالش و پاسخ‌ها به حدی زیاد باشد که با در دسترس بودن چند مورد از آن‌ها، امکان مدل‌سازی ریاضی به حمله‌کننده را ندهد، PUF قوی گفته می‌شود. برحسب همین دسته‌بندی می‌توان کاربرد معماری‌های PUF را به دودسته کلی تقسیم نمود. دسته‌ای که فقط یک جفت چالش و پاسخ دارد و جهت تولید کلید امن در تراشه‌ها کاربرد داشته و دسته دوم که تعداد چالش و پاسخ زیادی دارد و

جهت احرازهویت استفاده می‌گردد [29] ,[28] ,[2].

در مقاله [30] از نمونه کارهای پیاده‌سازی PUF و بررسی نتایج آن می‌باشد. در این کار طرح تولید رمز بر روی ASIC با تکنولوژی ۱۳۰ نانومتر صورت گرفته است. با توجه به این نتایج و خطای کمتر از ۰/۵۸ در میلیون، می‌توان به عملکرد خوب PUF و طراحی صورت گرفته پی برد. در این طراحی از انواع روش‌ها جهت تولید مدار PUF استفاده گردیده است. با بررسی پارامترهای PUF ازجمله یکتایی و تصادفی بودن، این پیاده‌سازی‌ها نیز به نتایج قابل‌قبول رسید.

به‌عنوان یکی از پرکاربردترین این پیاده‌سازی‌ها می‌توان به Arbiter PUF اشاره کرد که با استفاده از چندین مالتی‌پلکسر پشت سرهم و یک لچ به‌عنوان Arbiter در انتها ساخته می‌شود. پس از قرارگیری داده ورودی با توجه به تأخیر مسیرهای مختلف خروجی لچ صفر یا یک خواهد شد[25]، بهبود یکتایی پاسخ و قابلیت اطمینانPUF همواره موضوعی موردتوجه جهت پژوهش بوده که یکی از پژوهش‌ها در این زمینه مربوط به افزایش کاراییArbiter PUF می‌باشد که از گونه PUF های مبتنی بر مسیر تأخیر است و با استفاده از فیلیپ فلاپ بهبودیافته D صورت گرفته است [31]. همچنین مدلی دیگر از PUF بانام RF-PUF، که نگرشی نو در امر احرازهویت است معرفی‌شده است، در مقاله‌های[32] و [33] که جایزه مقاله برتر در کنفرانس معتبر HOST را در سال ۲۰۱۸ نیز دریافت کرده‌اند، البته مقاله [34] نیز به‌منظور بهبود کارایی این مدل‌ها ارائه شد.

در این پایانامه گوشه‌های تغییرات پروسه ساخت توسط نرم‌افزار Hspice بررسی و به وسیله مدل آماری مونته‌کارلو تاخیر مسیرهای تمامی PUFها محاسبه شده و شبیه‌سازی شده است که در فصل پنجم این پروسه به طور کامل شرح داده شده است.

باوجود به آنکه PUF توانسته است به میزان بسیار بالایی از نفوذ در سیستم‌های الکترونیکی جلوگیری کند، اما همچنان روش‌هایی برای عبور از دیوارهای امنیتی و حمله به سیستم وجود دارد، لذا با بررسی نقاط ضعف طراحی‌های مختلف و همچنین انجام حمله روی این طراحی‌ها می‌توان

پیشنهاداتی برای افزایش سطح امنیت و بهبود طراحی‌های جدید ارائه داد که در این پایان‌نامه قصد داریم با استفاده از روش‌های یادگیری ماشین مناسب جهت مدل‌سازی و حمله به PUF ها و بررسی آسیب‌پذیری آن‌ها، یک مدل مقاوم را ارائه دهیم.

۳-۲- PUF و حملات مدل‌سازی

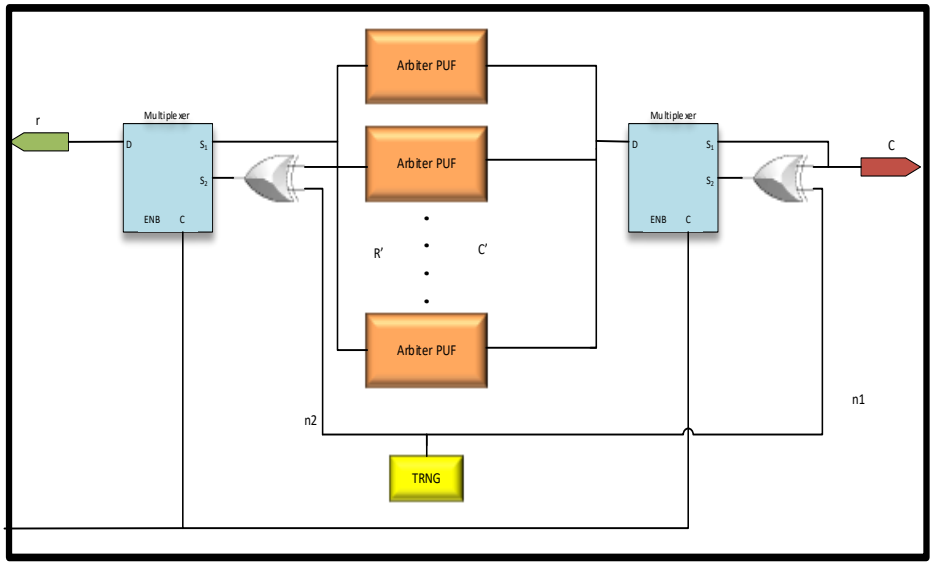
در یک رساله دکتری [35]، ۲۱ نوع از PUF ها ازنظر امنیتی موردبررسی قرار گرفتند که ازین تعداد فقط شش مدل از A-PUF مقاومت خوبی در برابر حملات یادگیری ماشین داشتند که از این بین، چهار مدل از پرقدرت‌ترین آن‌ها در ادامه معرفی و مورد بررسی بیشتر قرار خواهند گرفت.

مدل Slender PUF یکی از گونه‌های PUF است که در مقاله [36] معرفی شد، این مدل از بیت‌های تصادفی[1] استفاده می‌کند، به‌بیان‌دیگر اگر در یک Arbiter PUF اصلی به تعداد m بیت پاسخ وجود داشته باشد، خروجی را می‌توان به‌صورت $[r_1, r_2, \dots, r_m]$ نشان داد. این خروجی در Slender PUF به‌صورت $[t_1, \dots, t_k, r_\cdot, r_1, \dots, r_m, t_{k+1}, \dots, t_m]$ تغییر می‌کند که مقادیر t_1 تا t_m به‌صورت تصادفی تعیین می‌شوند که توسط یک RNG تولید می‌شوند، حال وقتی یک دشمن یک خروجی را دریافت می‌کند، نمی‌تواند بفهمد که کدام‌یک از بیت‌ها مربوط به پاسخ اصلی هستند. البته در مقاله [37] با استفاده از یک استراتژی تکاملی، موفق به شکستن امنیت این نوع PUF شده است، بنابراین این اقدامات برای جلوگیری از حملات مدلینگ یا حملات مبتنی بر ML کافی نبوده است.

یکی دیگر از پروتکل‌های مطرح‌شده برای مقابله با حملات مدل‌سازی، PolyPUF [38] نامیده می‌شود. ساختار این مدل در شکل (۳-۱) آورده شده است. این پروتکل برای مقابله با حملات ابتدا توسط یک RNG، عددی به‌صورت تصادفی تولید و با C ورودی این عدد را XOR می‌کند. مقدار C

[1] Random

جدید تولیدشده به n مدل مختلف از A-PUF اعمال می‌شود، (که n در اینجا تعداد بیت پاسخ است) پس از محاسبه پاسخ، دوباره تمام پاسخ‌ها با مقدار تصادفی جدید تولیدشده XOR می‌شوند. تعداد بیت‌های پیشنهادی برای عدد تصادفی ورودی ۲ بیت و برای خروجی ۳ بیت می‌باشد و طول داده Challenge پیشنهادی ۳۲ و ۶۴ بیت می‌باشد، ولی نویسندگان این مقاله درباره اینکه ۳ مضربی از ۳۲ یا ۶۴ نیست و چگونه باید با خروجی XOR شود اظهارنظری نکردند، لذا در تمامی مقالاتی که پیاده‌سازی این مدل را انجام داده‌اند، مقدار بیت تصادفی خروجی و ورودی برابر هم و برابر با مقدار ۲ بیت در نظر گرفته می‌شود.

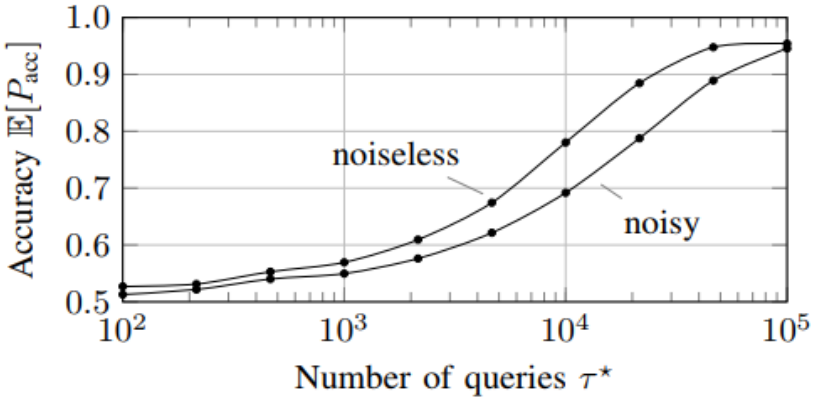


شکل (۳-۱) شمای Poly-PUF

به هرحال مقاله [39] توانست با استفاده از یک روش جدید به‌طور موفقیت‌آمیز به این مدل PUF حمله کند. این روش با استفاده از دسته‌بندی CRP ها قبل از حمله عمل می‌کند، به این صورت که اگر فاصله همینگ[1] بین ورودی‌های متوالی برابر ۱ باشد، با توجه به اینکه به دلیل وجود RNG در مدار به ازای هر ورودی این مدار می‌تواند چندین خروجی صحیح داشته باشد، پس درجایی که فاصله همینگ خروجی‌های متناظر کمترین شود، نشانه این است که مقدار عدد تصادفی XOR شده با

[1] Hamming Distance

ورودی و خروجی، در هر دو CRP ثابت مانده است. بدین‌صورت یک مجموعه از داده‌ها[1] تشکیل می‌شود، یعنی هنگام جمع‌آوری داده‌ها باید فاصله همینگ بین ورودی‌ها و خروجی انتخاب‌شده برابر با یک باشد. در ادامه توسط یک شبکه با یک نورون موفق به حمله با دقت بالای ۹۰ درصد به این مدل شده است. نمودار این حمله موفقیت‌آمیز در شکل (۳-۲) آورده شده است.



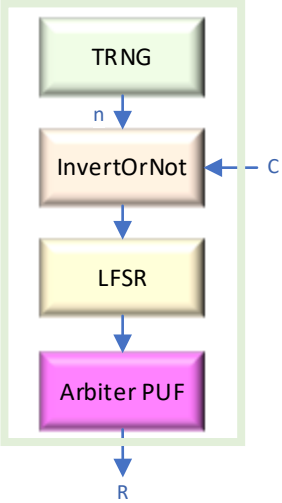
شکل (۳-۲) نمودار حمله به Poly-PUF [39]

یکی از بهترین مدل‌های ارائه‌شده برای مقابله با حملات RPUF نام دارد که در [40] ارائه‌شده است. این پروتکل می‌تواند در دو حالت کاری عمل کند، برای جلوگیری از حملات یادگیری ماشین ورودی دریافت شده را به‌صورت تصادفی عکس[2] می‌کند. در حالت کاری اول همه‌ی ورودی‌های دریافتی را عکس می‌کند و یا نمی‌کند که این عمل توسط یک RNG به‌صورت تصادفی انتخاب می‌شود. در حالت کاری بعدی برای قوی‌تر شدن مقاومت در برابر حملات RNG بجای تولید یک بیت عدد تصادفی، دو بیت تصادفی تولید می‌کند که این دو بیت می‌تواند چهار عملکرد مختلف داشته باشد: حالت اول: ورودی دریافت شده را بدون تغییر به خروجی انتقال می‌دهد، حالت دوم: تمام بیت‌های ورودی دریافت شده را عکس می‌کند و در دو حالت بعدی نیمه بالایی یا نیمه پایینی ورودی

[1] Dataset

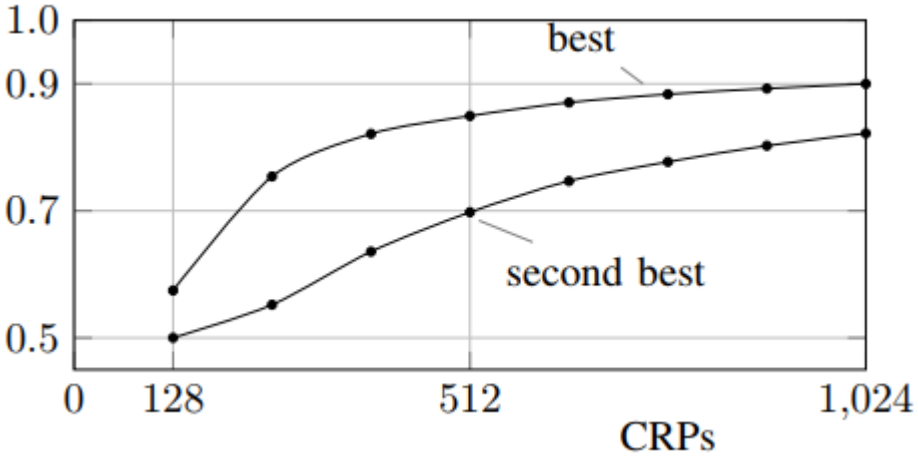
[2] Invert

دریافت شده را عکس می‌کند. درنهایت این ورودی تغییریافته را به یک LFSR[1] می‌دهد تا از این ورودی به تعداد بیت‌های پاسخ موردنظر زیرمجموعه تولید کند و با هر بار اعمال آن به یک A-PUF با یک بیت پاسخ می‌توان تعداد پاسخ بابیت‌های موردنظر را تولید کند. شمایی از این مدار در شکل (۳-۳) آورده شده است. نویسندگان ادعا کردند مدل ارائه‌شده نمی‌تواند با دقت بیشتر از حدود ۷۵٪ موردحمله قرار بگیرد. البته در مقاله [39] توسط یک استراتژی یادگیری جایگزین پس از آموزش مدل‌های مختلف، با استفاده از یک جستجوی طاقت‌فرسا توانست به‌دقت ۹۰ درصد برسد، نمودار این حمله در شکل (۳-۴) گزارش‌شده است. البته روش ارائه‌شده روش زمان‌بری می‌باشد، اما به‌هرحال امنیت این مدل را به‌صورت موفقیت‌آمیزی از بین برده است.



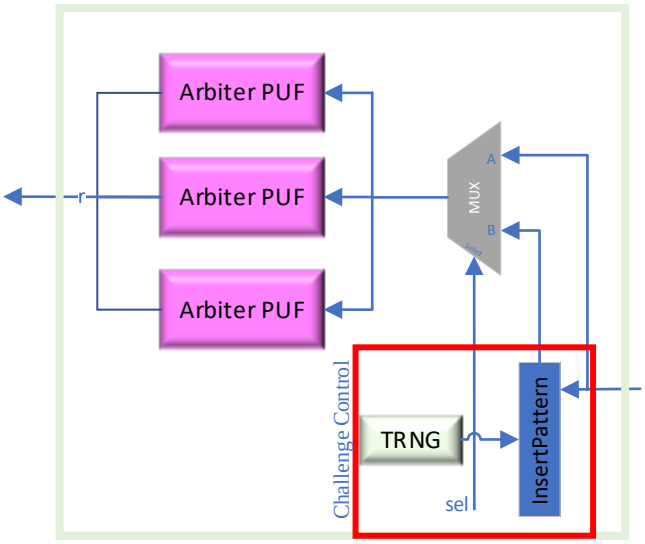
شکل (۳-۳) شمای کلی RPUF

[1] Linear-feedback Shift Register



شکل (۳-۴) نمودار حمله به RPUF [39]

در حیطه امنیت سخت‌افزار و PUF ها تا امروز مدل‌های مختلفی ارائه‌شده است که می‌توان گفت هرکدام کاربرد خود را دارند، اما در بسیاری موارد سعی بر کمتر شدن منابع مصرفی در کنار امنیت قابل‌قبول در برابر حملات الگوریتم‌های مختلف یادگیری ماشین بوده است، مدل OB-PUF که توسط مقاله [41] معرفی‌شده یکی از بهترین مدل‌های ارائه‌شده در این حوزه می‌باشد. این پروتکل به‌وسیله یک بخش بانام Challenge Control مقادیر چالش‌ها در ورودی را تغییر می‌دهد تا رابطه بین ورودی و خروجی را مخفی نماید تا از حملات یادگیری ماشین مصون بماند.



شکل (۳-۵) شمای کلی OB-PUF

در بخش کنترل‌کننده چالش[1]، یک RNG و یک بخش واردکننده الگو[2] وجود دارد. وظیفه این بخش اعمال دو الگو به‌صورت تصادفی به چالش ورودی می‌باشد، نویسندگان دو الگو با طول ۵ بیت پیشنهاد دادند که به‌صورت تصادفی به پنج بیت اول چالش یا پنج بیت آخر چالش اعمال می‌شوند. در این مطالعه تأکید شده که این الگوها باید بیشتری فاصله همینگ را از هم داشته باشند. همچنین نویسندگان طول داده ورودی را ۶۴ بیت و ۳ بیت خروجی را با استفاده از سه عدد A-PUF تولید می‌کنند. پس اگر داده ورودی را $[C_1, C_2, \dots, C_{64}]$ در نظر بگیریم داده ورودی پس از گذشتن از بخش کنترل‌کننده چالش به شکل $[C_1, C_2, \dots, C_{59}, 0, 1, 0, 1, 0]$ و یا $[1, 0, 1, 0, 1, C_6, C_7, \dots, C_{64}]$ تغییر می‌کند. با این کار ارتباط بین ورودی و خروجی دچار ابهام می‌شود، درواقع چالش‌هایی به وجود می‌آیند که چندین جواب متفاوت دارند.

سرور که از قبل توسط MUX موجود در این مدل به A-PUF ها دسترسی مستقیم داشته، یک مدل را نسبت به عملکرد این PUF ها آموزش داده است. حال با اعمال هر دو الگو به چالش

[1] Challenge Control

[2] Insert Pattern

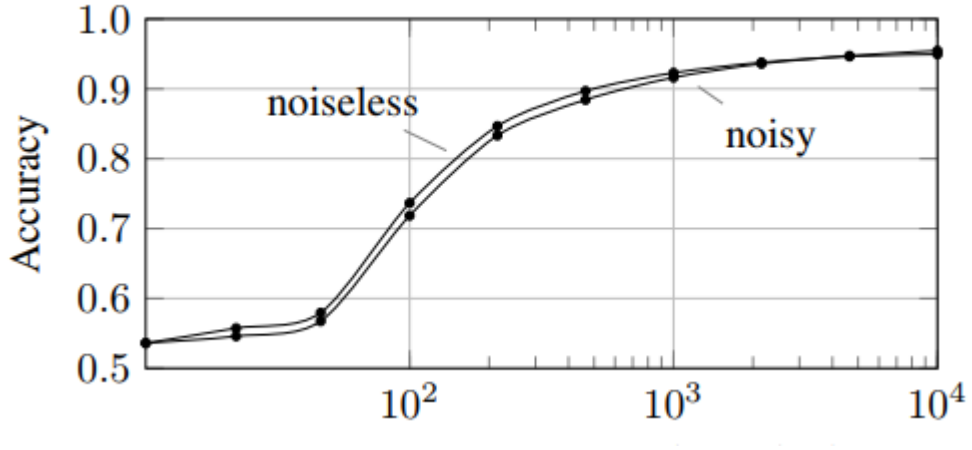
ورودی OB-PUF توسط مدل از قبل آموزش داده‌شده دو جواب محتمل به دست می‌آیند، حال پاسخ ارسالی از مدل باید حداقل به یکی از آن دو پاسخ پیش‌بینی‌شده شباهت داشته باشد تا احرازهویت انجام شود.

نتایج نویسندگان نشان داد که پس از جمع آوردی پاسخ‌های حاصل از چالش‌های تولیدشده به‌صورت تصادفی به تعداد $۱۰^۶$ داده، توسط الگوریتم رگرسیون لجستیک[1] نهایتاً به‌دقت بیشتر از ۷۲٪ دست پیدا کردند. البته در مسئله امنیت سخت‌افزار هر دقتی بیشتر از ۵۰٪ را می‌توان ناامن خواند.

باکمی بررسی این مدل می‌توان دریافت که این احتمال بسیار زیاد است که Arbiter PUF های استفاده‌شده در مدل مقادیر نامی[2] یکسانی برای همه پاسخ‌های مربوط به برخی چالش‌ها داشته باشد، حال اگر این مقادیر را برای دو حالت تغییریافته توسط هر دو الگو نسبت به یک چالش خاص پیدا کرد، گویی بخش کنترل‌کننده چالش را در این مدل بی‌اثر کردیم، به‌بیان‌دیگر در این مدل هر چالش می‌تواند دو پاسخ داشته باشد، چالش‌هایی که هر دو جواب آن‌ها یکسان باشد را انتخاب می‌کنیم، در مقاله [39] به‌وسیله جدا کردن این مقادیر و آموزش یک مدل با الگوریتم LR به‌دقت ۸۵٪ رسیده است، در مرحله بعدی با استفاده از این مدل به‌عنوان یک ابزار برای ابهام‌زدایی اقدام به جمع آموری داده‌های بیشتر بافاصله همینگ کم نسبت به خروجی پیش‌بینی‌شده نمود و به‌دقت بالای ۹۰٪ دست یافتند، البته می‌توان گفت این راه کمی طولانی بود. نمودار این حمله نیز در شکل (۳-۶) آورده شده است.

[1] Logistic Regression

[2] Nominal



شکل (۳-۶) نمودار حمله به مدل OB-PUF [39]

یکی از معیارهای مهم جهت تشخیص کیفیت معماری پیشنهادی، استفاده از حملات مدل‌سازی بر روی مدارات PUF است. بدین ترتیب که هدف مدل‌سازی ریاضی برحسب داشتن تعداد محدودی از جفت چالش و پاسخ‌ها است. این حملات زمینه‌های تحقیقاتی بسیاری به وجود آورده¬اند که از این جمله می‌توان به مرجع [42] اشاره کرد که حملات با استفاده از سه الگوریتم یادگیری ماشین، حملاتی به سه معماری PUF صورت گرفته و برحسب نتایج، نقاط قوت و ضعف معماری‌ها را نمایان کرده است. در [43]حملات با استفاده از ابزار Tensorflow و به زبان Python باقدرت بیشتری صورت گرفته است. همچنین با استفاده از بهینه‌سازی و اجرا در کارت گرافیکی (GPU) قدرتمند توانسته است معماری‌های پیچیده‌تر را نیز طی چند روز مدل‌سازی ریاضی نماید.

در ادامه به بررسی برخی از بهترین مقالات ارائه‌شده در این حوزه می‌پردازیم.

مقاله [44] با روش LR توانسته حمله‌ای رویArbiter PUF انجام دهد، داده‌های چالش-پاسخ خود را از سه روش شبیه‌سازی و با FPGA و ASIC به دست آورده است و در زمان ۲.۱ ثانیه با ۳۹۲۰۰ داده چالش-پاسخ روی مدل مربوطه با ۱۲۸ چالش و یک خروجی انجام دهد. همچنین توانسته بر روی XOR Arbiter PUF با ۱۲۸ چالش و پنج طبقه با یک پاسخ به‌وسیله ۵۰۰ هزار داده

چالش-پاسخ در زمان ۱۶:۳۶ ساعت حمله خود را انجام دهد. البته روی مدل Lightweight PUF با یک‌میلیون داده چالش-پاسخ و سایزی مشابه قبل به گفته مقاله ۲۶۷ روز زمان برای آموزش شبکه لازم است که منطقی نیست.

مقاله [45] از روش Active learning که یکی از متدهای یادگیری ماشین است که موفق به حمله به یک MUXPUF با ۶۴ چالش و یک پاسخ با داده‌های چالش-پاسخ کمتر نسبت به روش‌های دیگر شده است.

به گفته مقاله با این روش برای به دست آوردن خطای کمتر از ۴ درصد احتیاج به 2790 داده بوده که با این روش داده‌ها به ۸۱۱ عدد کاهش‌یافته است.

مقاله [46] مدلی بانام A-PUF روی FPGA ارائه کرده که با روش SVM به آن و مدل مشابه آن با ۵۰۰ هزار داده چالش-پاسخ حمله کرده است. این مدل توانسته دقت حمله را از ۹۸.۵ به ۵۰ درصد کاهش دهد و مقاوم باشد. لازم به ذکر است این مدل با الگوریتم‌های دیگر مورد ارزیابی قرار نگرفته است.

۳-۳- نتیجه‌گیری

با توجه به گسترش شبکه‌های اینترنت‌اشیا، نیاز به ارائه راهکارهای تضمین‌کننده ایمنی و امنیت ارتباطات امری ضروری است. با توجه به بررسی مطالعات گذشته پروتکل‌های قوی جهت احرازهویت، در این سیستم‌ها مطرح‌شده است، اما با پیشرفت این پروتکل‌ها استراتژی‌های حمله به آن‌ها نیز پیشرفت داشته است، لذا بررسی این حملات روی مدل‌های مختلف و انتخاب بهترین مدل از نظر امنیت و طراحی یک مدل امن جزو مطالعات مهم این حوزه است. در فصل ۴ روش پیشنهادی

خود را برای حمله به مدل‌های مقاوم بیان می‌کنیم و همچنین یک مدل با مقاومت بالاتر در برابر حملات مدل‌سازی و یادگیری ماشین ارائه می‌کنیم، سپس با بررسی و ارزیابی نتایج به دست‌آمده در بخش ۵ نتیجه‌گیری نهایی را انجام می‌دهیم.

فصل ۴: روش پیشنهادی

۴-۱- مقدمه

در این فصل ابتدا اقدام به طراحی یک شبکه کانولوشنی برای انجام حملات با دقت بالا به مدل‌های مختلف می‌کنیم، این به علت این است که بتوانیم از یک شبکه قوی برای حملات به انواع پروتکل‌ها استفاده کنیم، همان‌طور که در فصل ۲ نیز این شبکه‌ها معرفی شدند، قابلیت خوبی در مدل‌سازی سیستم‌های غیرخطی دارند که ازین‌رو ما ازین قابلیت استفاده خواهیم کرد.

در ادامه دو استراتژی حمله به دو مدل از PUF های قوی مطرح‌شده ارائه می‌کنیم، سپس با استفاده از مفاهیم فوق یک پروتکل مقاوم در برابر حملات یادگیری ماشین به نام SQ-PUF ارائه می‌دهیم.

۴-۲- شبکه کانولوشنی

با توجه به پیچیدگی ارتباط و وابستگی بین چالش و پاسخ در مدل‌های جدید، پس از بررسی روش‌های مختلف یادگیری ماشین مثل رگرسیون لجستیک، ماشین بردار پشتیبانی، جنگل تصادفی[1]، شبکه‌های عصبی مصنوعی[2] اعم از شبکه عصبی پیش‌خور[3] با چندین لایه پرسپترون، شبکه‌های مولد

[1] Random Forest

[2] Artificial Neural Network (ANN)

[3] Feedforward Neural Network

تخاصمی[1] و شبکه‌های عصبی کانولوشنی یا همگشتی[2] به این نتیجه رسیدیم که در بسیاری از موارد ازلحاظ سادگی یکی از بهترین روش‌ها برای حملات رگرسیون لجستیک می‌باشد ولی به دلیل خاصیت تک‌بعدی این روش در بسیاری از موارد که چندین پاسخ داریم، باید چندین شبکه آموزش بدهیم، از این‌رو ضمن بررسی حالت‌های مختلف با توجه به خاصیت غیرخطی، این مدل‌ها یکی از بهترین پیشنهادات برای پیش‌بینی مدل‌های غیرخطی شبکه‌های عصبی مصنوعی هستند. شبکه‌های عصبی پیش‌خور یکی از بهترین پیشنهادات برای این مدل‌سازی هستند. اگر بخواهیم این مدل‌سازی را پس از استخراج ویژگی‌ها استفاده کنیم، می‌توانیم از شبکه عصبی کانولوشنی (پیچشی) کمک بگیریم. در ادامه یک شبکه عصبی کانولوشنی که به‌صورت تجربی بهترین جواب‌ها را برای سایزهای مختلف و مدل‌های مختلف PUF پیش‌بینی کرده را ارائه دادیم که شمایی ازین شبکه در شکل (۴-۱) آمده است. برای آموزش این شبکه از روش بازگشت به عقب[3]، روشی برای محاسبه گرادیان‌ها استفاده نمودیم. تابع هزینه[4] استفاده‌شده برای آموزش این شبکه Binary Cross-Entropy می‌باشد. در آموزش شبکه کانولوشنی نگاه ما به مسئله یک مدل برای طبقه‌بندی[5] کردن چندین دسته می‌باشد. به‌عنوان تابع بهینه‌ساز[6] نیز از گرادیان کاهشی تصادفی[7] برای این مدل استفاده‌شده است. استفاده از شبکه کانولوشنی در این مطالعه و در نظر گرفتن مسئله به‌صورت مسئله طبقه‌بندی چند دسته‌ای یک کار جدید در این حوزه بود، اما این شبکه تنها درصورتی‌که مدل PUF بزرگ و پیچیده باشد، استفاده از یک شبکه کانولوشنی به‌صرفه است.

[1] Generative Adversarial Network (GAN)

[2] Convolutional Neural Network (CNN)

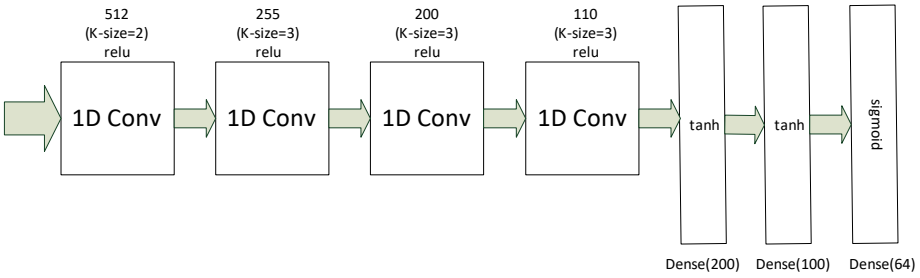
[3] Backpropagation

[4] Loss Function

[5] Classifier

[6] Optimizers

[7] Stochastic Gradient (SGD)



شکل (۴-۱) شمایی از شبکه کانولوشنی پیشنهادی

۴-۳- روش پیشنهادی حمله به OB-PUF

در این در این کار یک استراتژی آموزشی دیگر برای حمله به OB-PUF مطرح می‌شود. همان‌طور که در فصل قبلی ذکر شد، OB-PUF یکی از مقاوم‌ترین گونه‌های PUF ازنظر امنیت می‌باشد که این مدل توسط یک تکنیک قبلاً موردحمله قرارگرفته بود اما این تکنیک مشکلاتی اعم از پروسه طولانی داشت. در این مطالعه یک استراتژی دیگر با رویکرد طبقه‌بندی ارائه خواهیم داد.

۴-۳-۱- روند انجام

با توجه به این‌که تعداد بیت‌های پاسخ کم است، (مقدار پیشنهادی نویسندگان ۳ بیت بوده است) می‌توان آن را کدگذاری کرد، برای این منظور احتیاج به $۲^۳$ بیت داریم. مقدار پاسخ را به‌صورت یک عدد باینری در نظر بگیرید. با توجه به اینکه این مقدار ۳ بیت هست، همان‌طور که گفته شد Arbiter PUF-های استفاده‌شده در OB-PUF دارای مقادیر نامی یکسانی برای همه پاسخ‌های مربوط به برخی چالش‌ها داشته باشد، حال اگر این مقادیر را برای دو حالت تغییریافته توسط هر دو الگو نسبت به یک چالش خاص پیدا کنیم، گویی بخش کنترل‌کننده چالش را در این مدل بی‌اثر کردیم.

به‌بیان‌دیگر در این مدل، هر چالش می‌تواند دو پاسخ داشته باشد، چالش‌هایی که هر دو جواب آن‌ها یکسان باشد را انتخاب می‌کنیم، پس در اینجا مقادیر نامی یکسان برای همه پاسخ‌های مربوط به یک چالش جدا شدند، پس می‌توان مقداری بین ۰ تا ۷ را در نظر گرفت و به ازای هر داده در ۸ بیت آن را کد کرد یا به‌عبارتی‌دیگر مقدار موردنظر را به یک تغییر داد. در زیر نمونه‌ای از خروجی اصلی و خروجی کد شده آورده شده است.

جدول (۴-۱) مثال خروجی‌های کد شده در استراتژی ارائه شده

خروجی کد شده	خروجی اصلی
۰۰۰۰۰۱۰۰	۰۱۰
۰۰۰۰۰۰۰۱	۰۰۰
۰۰۱۰۰۰۰۰	۱۰۱

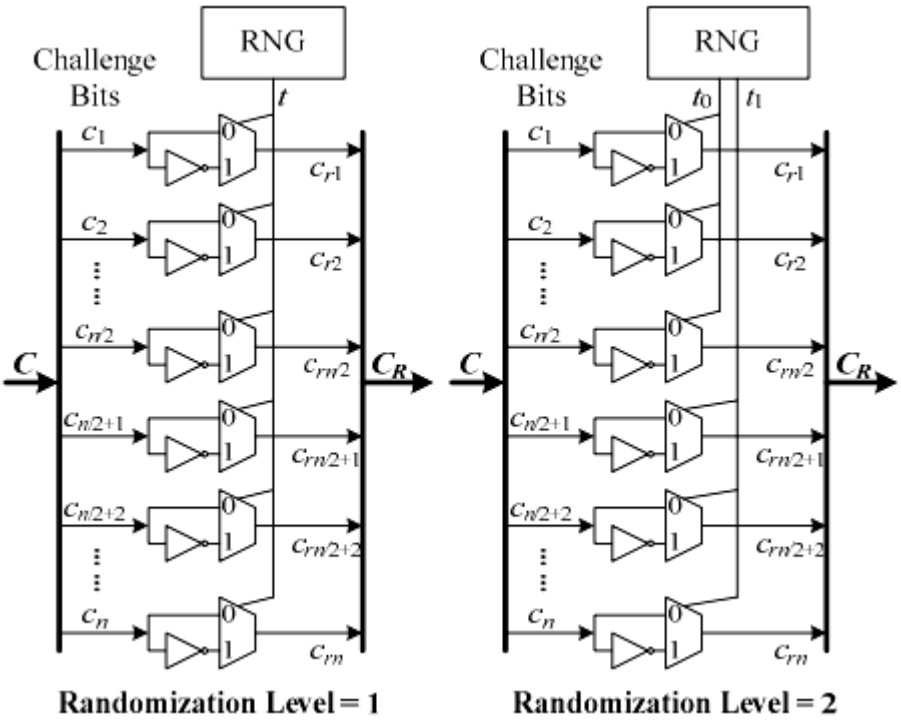
حال می‌توان با توجه به داده‌ها به این مسئله به‌عنوان یک مسئله طبقه‌بندی با ۸ دسته مختلف نگاه کرد. ما برای حل کردن این مسئله از شبکه کانولوشنی ارائه‌شده در بخش پیشین استفاده کردیم. در مقاله [39] در ابتدا فقط به یک ابزار ابهام‌زدا برای حل مسئله رسید اما با استفاده از کدگذاری ما می‌توانیم ارتباط بین ورودی و خروجی را به شبکه عصبی خود آموزش دهیم و در همان ابتدا به شبکه حمله کنیم، ازآنجاکه حتی می‌توانیم ازین شبکه نیز به‌عنوان یک ابزار ابهام‌زدای خیلی قوی نیز استفاده کنیم و در مرحله بعد به‌دقت بالاتر دست‌یابیم.

۴-۴- روش پیشنهادی حمله به RPUF

در این مطالعه یک استراتژی آموزشی دیگر برای حمله به R-PUF مطرح می‌شود. همان‌طور که در فصل قبلی ذکر شد این مدل توسط یک تکنیک قبلاً موردحمله قرارگرفته بود اما این تکنیک پروسه بسیار طولانی داشت در این مطالعه یک استراتژی دیگر ارائه خواهیم داد.

۴-۴-۱- روند انجام

همان‌طور که در فصل قبل ذکر شد این مدل PUF دارای دو حالت کاری می‌باشد که در حالت اول چالش ورودی به‌صورت سالم یا عکس شده به A-PUF ها اعمال می‌شد، در حالت کاری بعدی چهار عملکرد مختلف می‌تواند داشته باشد: حالت اول: ورودی دریافت شده را بدون تغییر به خروجی انتقال می‌دهد، حالت دوم: تمام بیت‌های ورودی دریافت شده را عکس می‌کند و در دو حالت بعدی نیمه بالایی یا نیمه پایینی ورودی دریافت شده را عکس می‌کند. پس وقتی یک استراتژی بتواند به حالت دوم این مدل حمله کند، پس آن روش می‌تواند به‌طریق‌اولی روی حالت اول نیز کارآمد باشد، در ادامه حالت دوم این مدل در نظر گرفته‌شده و استراتژی طبق این حالت بیان خواهد شد. بخش Challenge Randomization در این مدل در شکل (۴-۲) آورده شده تا مفهوم این استراتژی قابل‌درک باشد.



شکل (۴-۲) بخش Challenge Randomization در RPUF [40]

فرض کنید آرایه C یک چالش ورودی به مدار RPUF است، این چالش می‌تواند به یکی از چهار حالت زیر به A-PUF اعمال شود و به ازای هرکدام یک پاسخ مجزا دریافت نماید، پس هر چالش در حالت دوم می‌تواند تا چهار جواب درست داشته باشد.

$$[C_1, C_2, \dots, C_{30}, C_{31}, C_{32}, C_{33}, C_{34}, \dots, C_{62}, C_{63}, C_{64}] \rightarrow \begin{matrix} [C_1, C_2, \dots, C_{30}, C_{31}, C_{32}, C_{33}, C_{34}, \dots, C_{62}, C_{63}, C_{64}] \\ [\overline{C_1}, \overline{C_2}, \dots, \overline{C_{30}}, \overline{C_{31}}, \overline{C_{32}}, \overline{C_{33}}, \overline{C_{34}}, \dots, \overline{C_{62}}, \overline{C_{63}}, \overline{C_{64}}] \\ [\overline{C_1}, \overline{C_2}, \dots, \overline{C_{30}}, \overline{C_{31}}, \overline{C_{32}}, C_{33}, C_{34}, \dots, C_{62}, C_{63}, C_{64}] \\ [C_1, C_2, \dots, C_{30}, C_{31}, C_{32}, \overline{C_{33}}, \overline{C_{34}}, \dots, \overline{C_{62}}, \overline{C_{63}}, \overline{C_{64}}] \end{matrix}$$

هر چالش ورودی می‌تواند تا چهار پاسخ مختلف داشته باشد، حال اگر چهار چالش زیر را به RPUF اعمال کنیم، حتماً یکی از چهار پاسخ دریافتی بین تمامی این ورودی‌ها یکسان است:

$$\begin{matrix} Challenge_{0} \rightarrow [C_1, C_2, \dots, C_{20}, C_{31}, C_{32}, C_{33}, C_{34}, \dots, C_{62}, C_{63}, C_{64}] \\ Challenge_{1} \rightarrow [\overline{C_1}, C_2, \dots, \overline{C_{20}}, \overline{C_{31}}, \overline{C_{32}}, \overline{C_{33}}, \overline{C_{34}}, \dots, \overline{C_{62}}, \overline{C_{63}}, \overline{C_{64}}] \\ Challenge_{2} \rightarrow [\overline{C_1}, \overline{C_2}, \dots, \overline{C_{20}}, \overline{C_{31}}, \overline{C_{32}}, C_{33}, C_{34}, \dots, C_{62}, C_{63}, C_{64}] \\ Challenge_{3} \rightarrow [C_1, C_2, \dots, C_{20}, C_{31}, C_{32}, \overline{C_{33}}, \overline{C_{34}}, \dots, \overline{C_{62}}, \overline{C_{63}}, \overline{C_{64}}] \end{matrix} \rightarrow \begin{matrix} r_{01}, r_{02}, r_{03}, r_{04} \\ r_{11}, r_{12}, r_{13}, r_{14} \\ r_{21}, r_{22}, r_{23}, r_{24} \\ r_{31}, r_{32}, r_{33}, r_{34} \end{matrix}$$

پس در این استراتژی ابتدا یک چالش به مدل اعمال می‌کنیم و اعمال این چالش را تکرار می‌کنیم تا تمامی پاسخ‌های آن مشخص شود، پس‌ازآن عکس این چالش را به مدل می‌دهیم و پاسخ‌های آن را مقایسه می‌کنیم. در این حالت هر چهار پاسخ یکسان است با ترتیب غیر یکسان. این عمل را برای حالت‌های بعدی یعنی عکس نیمه پایینی چالش و عکس نیمه بالایی چالش نیز تکرار می‌کنیم، در مرحله بعد به همین صورت چهار چالش دیگر را به مدار اعمال می‌کنیم که فاصله همینگ بین چهار چالش اول و دوم یک باشد، حال پاسخی که بین پاسخ دسته اول و دوم کمترین فاصله همینگ را داشت انتخاب می‌شود. همچنین این هشت چالش که هر چهار چالش یک پاسخ یکسان دارند، باید به فهرستی اضافه شوند که دیگر تکرار نشوند، در ادامه چالش بعدی انتخابی باید فاصله همینگ یک نسبت به دسته دوم داشته باشد.

پس از طبقه‌بندی چالش‌ها و پاسخ‌ها می‌توانیم توسط یکی از روش‌های یادگیری ماشین به آن حمله موفق نماییم، در فصل بعد این حملات گزارش‌شده‌اند.

۴-۵- طراحی مدل PUF پیشنهادی مقاوم

همواره یکی از مهم‌ترین نکات در طرح‌های احرازهویت تولید کلید رمزنگاری تصادفی و امن است. به همین جهت از PUF که از ویژگی‌های فیزیکی ذاتی غیرقابل نمونه‌برداری استفاده می‌کند، به‌عنوان گامی مهم جهت آینده ایمن یاد می‌شود. از زمان ارائه PUF ها استراتژی‌های مختلفی برای حمله به آن‌ها مطرح شد که امنیت اکثر مدل‌های ارائه‌شده به‌کلی توسط برخی روش‌ها شکسته شد،

اما برخی مدل‌ها مقاومت بالاتری از خود نشان دادند. در فصل قبل مدل‌های متفاوتی که جزو امن‌ترین مدل‌های ارائه‌شده بودند آورده شد و امنیت آن‌ها موردبررسی قرار گرفت. در بخش قبل نیز دو استراتژی جدید برای حمله به دو مدل که از امن‌ترین مدل‌ها می‌باشند ارائه شد. می‌توان گفت با پیشرفت یادگیری ماشین اهمیت بالا رفتن امنیت در تمام حوزه‌ها دوچندان می‌شود، این‌رو پس از بررسی مدل‌های مختلف ازنظر امنیت ضمن استفاده از مزایای آن‌ها و در نظر گرفتن معایب اقدام به طراحی یک مدل با امنیت بالاتر نمودیم که در ادامه به بررسی این مدل خواهیم پرداخت.

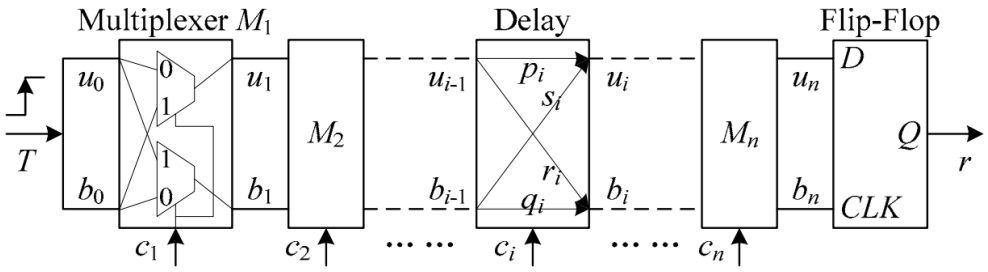
قبل از هر چیز باید ابتدا یک مدل از A-PUF اصلی ارائه‌شده را بشناسیم و سپس اقدام به ایمن‌سازی این PUF نماییم پس‌ازآن باید روندی برای بازیابی پاسخ ارائه دهیم که سرور بتواند تشخیص دهد که آیا دستگاه اصلی پاسخ را فرستاده یا یک دشمن تا عمل احرازهویت به‌درستی صورت گیرد.

۴-۵-۱- روند انجام

یک A-PUF می‌تواند 2^n چالش-پاسخ مختلف داشته باشد که در اینجا n تعداد بیت‌های چالش است. به شکل (۴-۳) توجه کنید، پس از قرارگیری هر چالش T از طریق MUX ها از دو مسیر مختلف به داور[1] می‌رسد، اگر مسیر D زودتر برسد داور خروجی را یک می‌کند در غیر این صورت خروجی صفر می‌شود که این عمل همبستگی[2] بالایی ایجاد می‌کند که می‌تواند توسط الگوریتم‌های یادگیری ماشین استفاده شود.

[1] Arbiter

[2] Correlation



شکل (۴-۳) شمای کلی از Arbiter PUF

همان‌طور که در شکل (۴-۳) نشان داده‌شده ما از p_i, q_i, s_i, r_i برای نمایش تأخیر مسیرهای مالتی‌پلکسر M_i استفاده می‌کنیم، حال وقتی یک چالش-پاسخ معین به این A-PUF اعمال نماییم، پاسخ r از طریق فرمول زیر محاسبه خواهد شد:

$$r = \Theta(\vec{W} \times \vec{\Phi}) \qquad \Theta(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

$$\vec{W} = (w_1, \ldots, w_{n+1}) \qquad w_i = \frac{\delta_{i-1}^0 + \delta_{i-1}^1 + \delta_i^0 - \delta_i^1}{2} \qquad \begin{matrix} \delta_i^0 = q_i - p_i \\ \delta_i^1 = r_i - s_i \end{matrix}$$

$$\vec{\Phi} = (\phi_1, \ldots, \phi_n, 1)^T \qquad \phi_l = \prod_{i=l}^{n} (1 - 2c_i)$$

در اینجا $\vec{\phi}$ و r با استفاده از یک چالش معین تعیین می‌شوند، پس بردار $\vec{W}$ با استفاده از $\vec{W} \times \vec{\phi} = 0$ می‌تواند یک ابر صفحه[1] جداکننده را تعیین کند که چالش‌هایی که پاسخ آن‌ها صفر می‌شوند را از دیگر چالش‌ها جدا سازد، از همین طریق روش‌های یادگیری ماشین بعد از پیدا کردن این ابر صفحه می‌توانند چالش‌ها را دسته‌بندی نمایند. ازین رو در روش‌های مختلف جهت مقاوم‌سازی PUF ها ارتباط بین چالش پاسخ را قطع کردند، در روش‌هایی مثل XOR-PUF پاسخ را مخفی و در روش‌هایی مثل RPUF و OB-PUF چالش را مخفی ساختند، اما به‌هرحال توسط استراتژی‌هایی

[1] Hyper Plane

توانستند به این ابر صفحه دست یابند.

در اینجا نوع دیگری از PUF ها را معرفی می‌کنیم که با استفاده از وابستگی زمانی، چالش و پاسخ را مخفی می‌سازد و ازین طریق امنیت را در سیستم‌های الکترونیکی بهبود می‌بخشد.

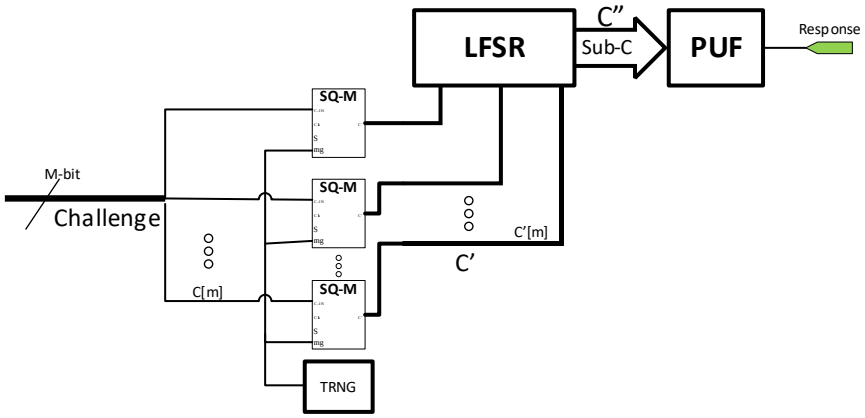
هدف ما از ارائه این مدل جلوگیری از یادگرفتن ارتباط بین چالش‌ها و پاسخ‌ها و پنهان کردن ابر صفحه از طریق روش‌های یادگیری ماشین و مقاوم‌سازی در برابر حملات مدل‌سازی می‌باشد، پس از مقاوم‌سازی هنوز هم سرور می‌تواند با استفاده از یک روش بازیابی عملیات احرازهویت را انجام دهد.

برخلاف Arbiter PUF های اصلی که فقط یک بیت برای پاسخ دارند، در بسیاری از کاربردها به چندین بیت پاسخ نیاز است، برای این مسئله دو راه‌حل وجود دارد: [47][4] اولین راه این است که مانند مدل‌هایی مثل OBPUF یا PolyPUF از N مدل مختلف Arbiter PUF برای تولید N پاسخ مختلف استفاده کنیم، اما با توجه به اینکه در بسیاری از کاربردها مانند احرازهویت در سیستم‌های IOT نیازمند یک سخت‌افزار با منابع مصرفی کم هستیم، ازاین‌رو ما در این مطالعه با استفاده از روش دوم یعنی استفاده از یک LFSR[1] و یک Arbiter PUF طراحی خود را انجام دادیم. در این روش با استفاده از یک LFSR از چالش ورودی به تعداد بیت موردنیاز پاسخ، زیرمجموعه چالش[2] تولید و به A-puf اعمال می‌کنیم.

در شکل (۴-۴) یک مدل از SQ-PUF با m بیت چالش نشان داده‌شده است، در این روش می‌توانیم ماژول‌های SQ را به همه (۱۰۰٪) یا فقط به بخشی از چالش‌ها تخصیص دهیم. پس از تغییر چالش‌ها به‌وسیله ماژول‌های SQ، آن‌ها به یک LFSR اعمال می‌شود تا زیرمجموعه‌هایی از این چالش برای تولید پاسخ‌های مختلف تولید شوند.

[1] Linear-feedback Rhift Register

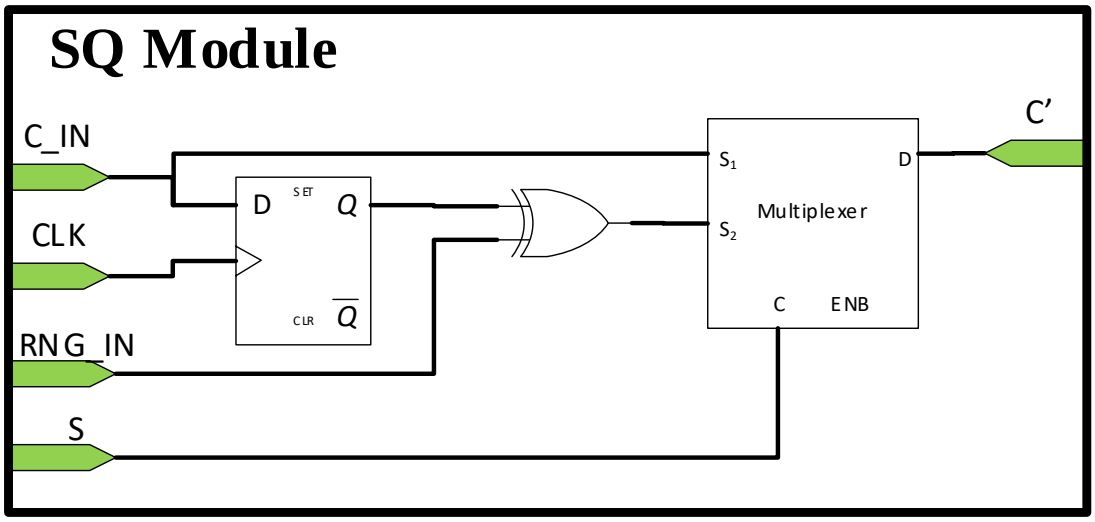
[2] Sub-challenge



شکل (۴-۴) شمای کلی SQ-PUF پیشنهادی

ماژول SQ

ایده اصلی SQ-PUF وابسته کردن دو چالش متوالی به یکدیگر است، به‌عبارت‌دیگر در این مدل PUF، پاسخ متأثر از دو چیز است، یکی ترتیب وارد شدن چالش‌ها و یکی عدد تصادفی تولیدشده توسط RNG می‌باشد. این ماژول می‌تواند به همه و یا فقط به برخی از چالش‌ها اعمال شود، این چالش‌ها می‌توانند در زمان ساخت این PUF به‌صورت تصادفی انتخاب شوند. به‌صرف وجود یک RNG یک بیتی، این مدار خاصیت تصادفی بودن خود را حفظ می‌کند، ازین‌رو هر پاسخ هر چالش با توجه به چالش قبلی دارای دو جواب صحیح می‌باشد. ازآنجاکه قبلاً کارهای زیادی برای طراحی RNG انجام‌شده برای مثال کارهایی مثل [53][52][51][50][49][48] و ...، در اینجا هدف ما طراحی یک مدل مقاوم در برابر حملات مدل‌سازی است، پس فرض می‌کنیم یکRNG شایسته، بهینه و با پاسخ‌دهی دقیقاً ۵۰/۵۰ در این مدل استفاده شده است.



شکل (۴-۵) ماژول SQ در مدل SQ-PUF پیشنهادی

در ماژول SQ که در شکل (۴-۵) آمده، با استفاده از یک D فلیپ-فلاپ چالش را به حالت قبلی وابسته می‌کنیم، حالت اولیه را می‌توان صفر در نظر بگیریم، در این بخش از یک XOR جهت تصادفی سازی چالش بهره می‌بریم به صورتی که اگر عدد تصادفی تولیدی توسط تولیدکننده عدد تصادفی برابر یک باشد چالش عکس می‌شود، در غیر این صورت خودش به خروجی انتقال می‌یابد. همچنین در این ماژول به کمک یک مالتی‌پلکسر که فقط برای بار اول به آن دسترسی داریم ارتباط مستقیمی بین چالش ورودی و PUF برقرار می‌شود، یعنی به‌واسطه این مالتی‌پلکسر می‌توان با نادیده گرفتن ماژول SQ به هسته PUF دسترسی داشت، این به‌منظور آموزش سرور برای بار اول تعیین‌شده است.

۴-۵-۲- کارکرد مدل SQ-PUF پیشنهادی

همان‌طور که گفته شد در SQ-PUF هر پاسخ وابسته به چالش خود و چالش قبلی خود است، همچنین هر چالش می‌تواند دو پاسخ متفاوت صحیح داشته باشد.

به‌عبارت دیگر اگر C_t و C_{t-1} ثابت باشد، C_t دو جواب صحیح خواهد داشت که این ناشی از وجود RNG در مدار است. در ادامه مراحل ثبت‌نام و احرازهویت به ترتیب موردبررسی دقیق‌تر قرار خواهند گرفت.

مرحله ثبت‌نام[1]

مراحل قسمت ثبت‌نام فقط یک‌بار جهت آموزش رفتار PUF موردنظر برای احرازهویت به سرور انجام می‌شود، در این قسمت سرور می‌تواند به بخش‌های خاصی نظیر مالتی‌پلکسرهای داخلی دسترسی داشته باشد، اما پس از اتمام کار این دسترسی‌ها از بین می‌رود. در شکل (۴-۶) الگوریتم ۱ مراحل این بخش آورده شده است. بر اساس این الگوریتم ابتدا سرور، تعداد N چالش را به‌صورت تصادفی تولید و به PUF ارسال می‌کند، تعداد این چالش‌ها باید به‌اندازه کافی بزرگ باشد، SQ-PUF در این فاز توسط مالتی‌پلکسرهای موجود در ماژول SQ دسترسی مستقیم چالش‌ها به Arbiter PUF را فراهم می‌کند، سپس از هر چالش ورودی α بار زیرمجموعه‌های این چالش توسط LFSR تولید و به Arbiter PUF برای تولید α بیت پاسخ اعمال می‌شود، پس‌ازاین مرحله دسترسی مستقیم به Arbiter PUF توسط مالتی‌پلکسرها غیرفعال می‌شود و چالش‌های اعمال ازین پس باید حتماً از ماژول‌های SQ عبور کنند تا به LFSR برسند. در ادامه پاسخ‌های تولیدشده به سرور ارسال می‌شوند، حال در سرور دسته‌ای از چالش-پاسخ‌های معین داریم که مستقیماً توسط A-PUF تولیدشده‌اند، پس سرور می‌تواند توسط این دسته که شامل N عدد CRP است یک مدل آموزش دهد تا این مدل ابر صفحه جداکننده پاسخ‌ها را به ازای چالش‌ها با دقت بالایی پیدا کند یا به‌عبارت‌دیگر وابستگی بین چالش پاسخ را با دقت بالایی یاد بگیرد.

[1] Enrollment

Algorithm 1 Enrollment ▷ Server

1: ▷ $Create\ Challenge \leftarrow TRNG(N)$
2: $C_{(1,2,\ldots,N)} \leftarrow Server$
3: $\mathcal{A}$ccess to C
4: **for** $i = 1, 2, \ldots, N$ **do**
5: $\quad C'_{(1,2,\ldots,\alpha)} = LFSR(C_i)$
6: $\quad$ **for** $j = 1, 2, \ldots, \alpha$ **do**
7: $\quad\quad \mathbf{r}_{(i,j)} = \mathrm{ArbiterPUF}(\mathcal{C}'_{\mathrm{j}})$
8: $\quad$ **end for**
9: **end for**
10: $\mathbf{r}_{(i,j)} \rightarrow Server$
11: $\mathcal{D}$issable Access to C
12: ▷ TrainModel With CRPs

شکل (۴-۶) الگوریتم ۱ مرحله ثبت‌نام SQ-PUF

مراحل بخش احراز هویت[1]

در شکل (۴-۷) الگوریتم ۲ مراحل بخش احرازهویت مطرح‌شده است، این بخش هر موقع و برای همیشه می‌تواند انجام شود، درواقع هر بار سرور نیاز به احرازهویت داشته باشد توسط طی کردن این مراحل می‌تواند روند احرازهویت را به‌خوبی انجام دهد، همچنین این مراحل نشان‌دهنده آن است که یک‌روند بازیابی قابل‌اجرا برای مدل SQ-PUF ارائه‌شده است. در این بخش هیچ دسترسی به مالتی‌پلکسرهای داخل ماژول SQ وجود ندارد، در ابتدا یک چالش به‌صورت تصادفی توسط سرور تولیدشده و به مدل PUF ارائه‌شده ارسال می‌شود، پس از دریافت چالش، بیت‌هایی که ماژول SQ به آن‌ها اضافه‌شده وارد ماژول شده و به D فیلیپ‌فلاپ‌های داخل ماژول اعمال می‌شوند، مابقی بیت‌ها مستقیماً اعمال می‌شوند. این عمل باعث می‌شود که این PUF به چالش خود و قبلی وابستگی پیدا کند، سپس با توجه به حالت قبلی فلیپ‌فلاپ‌ها که برای بار اول آن را صفر فرض می‌کنیم، هرکدام از خروجی‌های جدید تولید می‌شود بانام ’C با عدد تصادفی تولیدشده توسط RNG موجود که ۰ یا ۱ است XOR می‌شوند، خروجی ”C حاصل به LFSR اعمال می‌شود. باید توجه داشت فقط این عملیات

[1] Authentication

برای بیت‌هایی که ماژول SQ به آن‌ها اضافه‌شده انجام می‌شود، مابقی بیت‌ها مستقیماً به LFSR می‌رسند، به‌عنوان‌مثال ۸۰٪ بیت‌ها برای بار اول به‌صورت تصادفی انتخاب و ماژول‌های SQ سر راه آن‌ها می‌توانند قرار گیرند. پس‌ازاین مرحله از "C به تعداد بیت‌های موردنیاز پاسخ زیرمجموعه توسط LFSR تولید و هر بار به A-PUF داده می‌شود و پاسخ تولیدی برای سرور ارسال می‌شود.

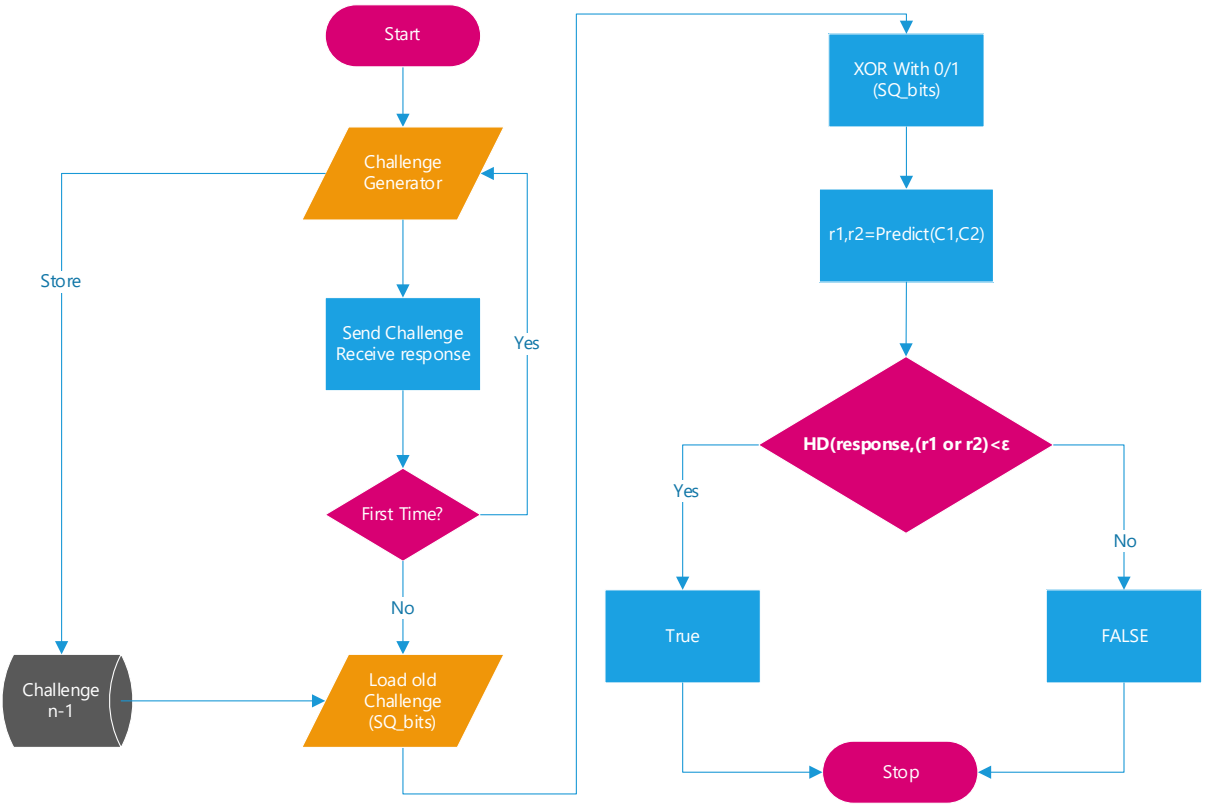
در بخش سرور اگر ماژول‌های SQ در SQ-PUF موردنظر فقط به برخی چالش‌ها اعمال‌شده باشند، اینجا هم دقیقاً باید D فلیپ‌فلاپ‌های موجود در سرور دقیقاً به همان بیت‌های چالش اعمال شوند. پس از اعمال چالش به D فلیپ‌فلاپ‌ها خروجی آن را یک‌بار با ۰ و یک‌بار با ۱، XOR می‌کنیم، سپس پاسخ هر دو خروجی تولیدشده، توسط مدلی که در فاز ثبت‌نام آموزش داده‌شده بود پیش‌بینی می‌شوند. در آخر حتماً باید پاسخ دریافت شده از SQ-PUF حداقل معادل یکی از پاسخ‌های پیش‌بینی‌شده باشد که این مرحله را با محاسبه فاصله همینگ بین آن‌ها بررسی می‌کنیم. در اینجا ε برابر صفر یا یک مقدار کم که با توجه به نویز موجود می‌توان آن را محاسبه نمود، است. لازم به ذکر است با توجه به وابستگی خروجی SQ-PUF به ترتیب چالش‌ها، اولین پاسخ را برابر غیر معتبر در نظر می‌گیریم و از آن به بعد احرازهویت را انجام می‌دهیم، با این عمل وابستگی بین چالش-پاسخ تا حد زیادی از بین رفته و ارتباط بین آن‌ها غیرقابل‌پیش‌بینی می‌شود.

Algorithm 2 Authentication (∞ times)

1: ▷ $Create\ Challenge \leftarrow TRNG()$
2: $C \leftarrow Server$
3: $\mathbf{n} = \mathrm{TRNG}()$
4: $C^{'} = D_{FF}(C)$
5: $C^{"} = C^{'} \mathcal{XOR}\ \mathbf{n}$
6: $C^{\gamma}{}_{(1,2,\ldots,\alpha)} = LFSR(C^{"})$
7: **for** $j = 1, 2, \ldots, \alpha$ **do**
8: $\quad \mathbf{r}_{(j)} = \mathrm{ArbiterPUF}(\mathcal{C}^{\gamma}{}_{j})$
9: **end for**
10: $\mathbf{r}_{(j)} \rightarrow Server$
▷ $Server$:
11: $Create\ Challenge \rightarrow TRNG() \rightarrow PUF$
12: Recive $\mathbf{r} \leftarrow PUF$
13: **if** $FirstTime$ **then** $C_{old} \leftarrow Save\ Challenge$ ▷ $NotValid$
14: **else**
15: $\quad C^{'} = D_{FF}(C)$
16: $\quad$ **for** $\mathbf{n} = 0, 1$ **do**
17: $\quad\quad C^{"}_{\mathbf{n}} = C^{'} \mathcal{XOR}\ \mathbf{n}$
18: $\quad\quad r_{\mathbf{n}} = Predict(C^{"}_{\mathbf{n}})$
19: $\quad$ **end for**
20: $\quad h = \min(\mathbf{HD}(\ \mathbf{r}, r_{\mathbf{0}}), \mathbf{HD}(\ \mathbf{r}, r_{\mathbf{1}}))$
21: $\quad$ **if** $h > \epsilon$ **then** $Reject$
22: $\quad$ **end if**
23: $\quad C_{old} \leftarrow Save\ Challenge$
24: **end if**

شکل (۴-۷) الگوریتم ۲ مرحله احراز هویت SQ-PUF

معمولاً در سرور سربار سخت‌افزاری مطرح نیست اما با توجه به ذخیره چالش قبلی و بیت‌هایی که به آن‌ها ماژول SQ اضافه‌شده است، می‌توان از بیت‌های قبلی بجای استفاده از D فلیپ‌فلاپ استفاده نمود و این سربار را به‌شدت کاهش داد. در شکل (۴-۸) مراحل سرور به‌صورت یک فلوچارت برای درک بهتر عملکرد آن آورده شده است.



شکل (۴-۸) فلوچارت عملکرد سرور در بخش احرازهویت در SQ-PUF

۴-۶- نتیجه‌گیری

یکی از مهم‌ترین چالش‌های امروزی بحث امنیت، مقاومت در مقابل روش‌های یادگیری ماشین است. این امنیت زمانی که اطلاعات محرمانه‌ای از سیستمی به سیستم دیگر انتقال پیدا می‌کند دارای اهمیتی دوچندان است. یکی از بهترین روش‌های برقراری امنیت احرازهویت قبل از ارسال و دریافت اطلاعات است. از پرکاربردترین روش‌های احرازهویت استفاده از سیستم‌های مبتنی بر PUF است که با توجه به خواص فیزیکی که حین ساخت تعیین می‌شوند، این عمل را آسان می‌سازد. ازین‌رو با توجه به پیشرفت روزافزون روش‌های یادگیری ماشین، احتیاج به پیشرفت این سیستم‌ها نیز نیاز است. در این فصل ضمن ارائه دو روش جدید برای حمله به دو نوع PUF مقاوم‌شده در برابر حملات، اقدام به

طراحی یک مدل امن در برابر این‌گونه حملات نمودیم. این مدل جدید امن بر پایه وابسته‌سازی چالش‌ها به ترتیب و اضافه کردن خاصیت تصادفی بودن به آن‌ها سعی در مخفی کردن رابطه بین چالش-پاسخ از روش‌های یادگیری ماشین را دارد. در فصل بعد به بررسی و آنالیز داده‌های به‌دست‌آمده از این روش‌ها می‌پردازیم.

فصل ۵: نتایج و پیشنهادها

۵-۱- مقدمه

در این فصل به بررسی مدل ارائه‌شده و استراتژی‌های مطرح‌شده در فصل ۴ می‌پردازیم. در ابتدا باید روشی برای شبیه‌سازی تغییرات فرآیند ساخت و آنالیز آن ارائه کنیم، سپس به بررسی دو استراتژی مطرح‌شده برای حمله به دو مدل PUF مقاوم می‌پردازیم، پس از آن با بررسی دقیق مدل ارائه شده مقاومت مدل خود را اثبات می‌کنیم.

۵-۲- آنالیز تغییرات ساخت

برای انجام شبیه‌سازی‌ها در این مطالعه احتیاج به پیدا کردن محدوده مقدار تأخیر مسیرها در PUF می‌باشد. یکی از روش‌های موجود برای بررسی تغییرات ساخت، استفاده از شبیه‌سازی‌های پیاپی به‌صورت تصادفی با روش مونت‌کارلو[1] در محدوده گوشه‌های[2] مدار می‌باشد، لذا به‌وسیله پیاده‌سازی یک Arbiter PUF با ۶۴ چالش ورودی و یک پاسخ در نرم‌افزار Hspice به‌وسیله الگوریتم مونت‌کارلو و مقادیرگوشه‌های بدست آمده مدار، آن را ازنظر تأثیر فرآیند ساخت بر تأخیر مسیرهای مختلف در ۱۰۰۰ نمونه مختلف مورد بررسی قراردادیم و مقادیر میانگین ۵۲.۳ پیکوثانیه و واریانس ۱۱ پیکو ثانیه در بدترین حالت به دست آمد که در مدل‌سازی‌ها مقادیر تأخیر مسیرها با یک توزیع

[1] Monte Carlo

[2] Corner

نرمال تعیین شدند تا نزدیک‌ترین حالت به واقعیت مورد شبیه‌سازی قرار بگیرد.

جدول (۵-۱) مقادیر بدست آمده برای محدوده تغییرات تاخیر مسیرها در فرآیند ساخت

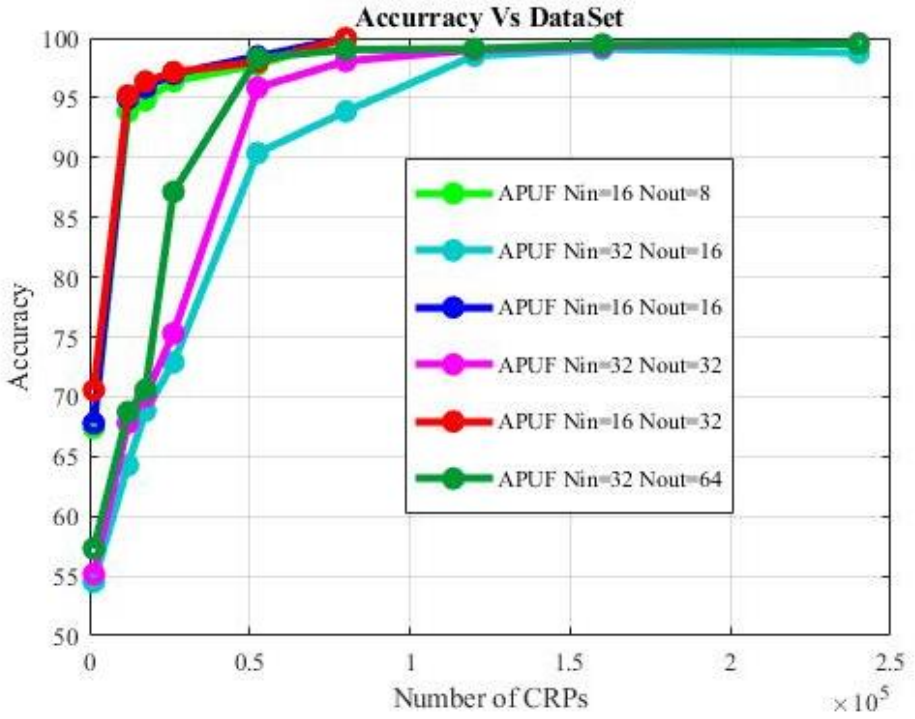
mean	52.3 ps
variance	11 ps

۵-۳- بررسی حملات

در این بخش با توجه به استراتژی‌های مطرح شده در فصل چهارم به مدل‌های OB-PUF و R-PUF حمله و میزان مقاومت این مدل‌ها و میزان تأثیر استراتژی حمله را موردبررسی قرار می‌دهیم.

۵-۳-۱- حمله به Arbiter PUF

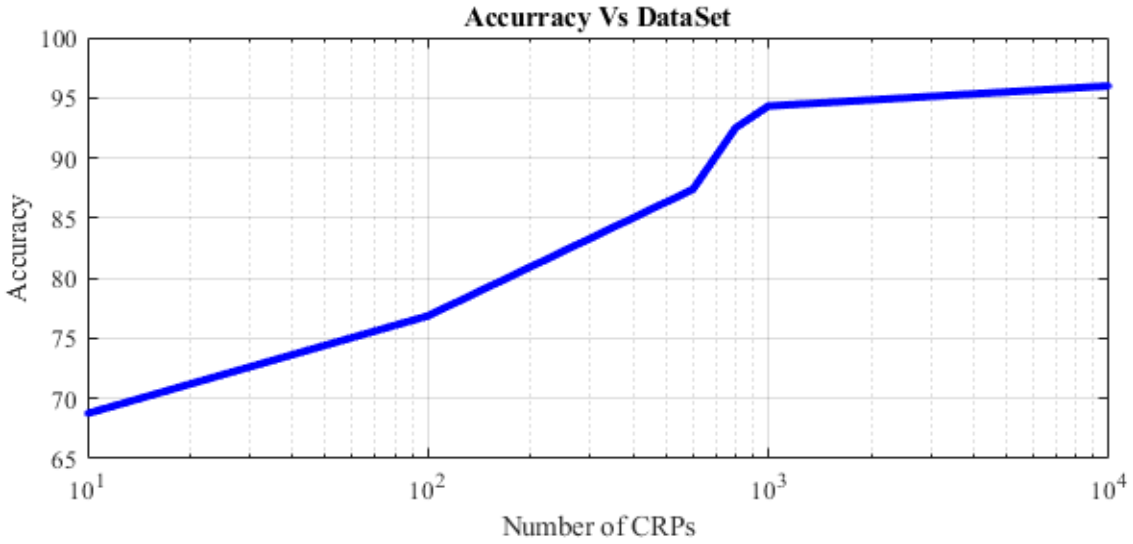
یک مدل Arbiter PUF اصلی با سایزهای مختلفی از چالش-پاسخ توسط شبکه کانولوشنی که قبلاً ارائه شد موردحمله قرار گرفت تا ضمن بررسی تأثیر سایزهای مختلف چالش-پاسخ از قابلیت شبکه کانولوشن در پیدا کردن ابر صفحه مربوط به PUF با سایزهای مختلف اطمینان حاصل کنیم. طبق نمودار ارائه‌شده در شکل (۵-۱) سایز پاسخ تأثیر چندانی در دقت حمله در روش شبکه کانولوشن ندارد، اما در روش‌هایی مثل رگرسیون لجستیک باید به ازای هر پاسخ یک شبکه جداگانه آموزش داده شود، اما تعداد چالش‌ها یک عامل محدودکننده دقت در برابر تعداد چالش-پاسخ‌ها می‌باشد، یعنی هرچه تعداد بیت چالش بیشتر باشد به تعداد داده‌های بیشتری برای حمله با دقت ٪۹۹.۹ نیاز داریم. طبق نمودار ارائه‌شده، روش شبکه کانولوشن قابلیت حمله موفق با دقت ٪۹۹.۹ به سایزهای مختلف PUF دارد.



شکل (۵-۱) نمودار حمله به Arbiter PUF با شبکه CNN پیشنهادی

۵-۳-۲- حمله به OB-PUF با استراتژی جدید

با استفاده از استراتژی یادگیری مطرح‌شده در فصل قبل، برای حمله به مدل OB-PUF، با جمع‌آوری $۱۰^۴$ داده و استفاده از شبکه کانولوشن ارائه‌شده در فصل قبل موفق شدیم تا در این مدل PUF به‌دقت بالای ۹۵ درصدی دست‌یابیم. این روش جدید ارائه‌شده می‌تواند ضمن داشتن دقت بالا، با کمترین زمان ممکن نسبت به روش‌های قبلی به مدل OB-PUF حمله کرد که این نشان‌دهنده این است که این مدل PUF نتوانسته در مقابل روش‌های یادگیری ماشین از خود مقاومت خوبی نشان دهد. می‌توانید در شکل (۵-۲) نمودار مربوط به این حمله را مشاهده نمایید.



شکل (۵-۲) نمودار حمله به OB-PUF با استفاده از شبکه CNN و استراتژی ارائه شده

۵-۳-۳- حمله به RPUF

با استفاده از استراتژی یادگیری حمله به مدل RPUF که در فصل قبلی مطرح شد، با جمع‌آوری $۱۰^۴$ داده و استفاده از شبکه کانولوشن ارائه‌شده در فصل قبل این حمله را انجام دادیم.

موفق شدیم تا حمله موفق با دقت ۸۳٪ داشته باشیم، این حمله به نوع دوم این مدل که دارای امنیت بالاتری نسبت به مدل اول بود انجام‌شده است، این حمله نسبت به روش ارائه‌شده در مقاله [39] دارای دقت کمتری است اما بسیار ساده‌تر و زمان کمتر نسبت به آن روش می‌توان این عمل را انجام داد. می‌توان گفت این مدل PUF به نسبت دیگر مدل‌ها از مقاومت بالاتری برخوردار است، اما در بسیاری از حوزه‌ها هر عددی بیشتر از ۵۰٪ برای حمله نشان‌دهنده ضعف در آن روش است.

۵-۴- بررسی SQ-PUF

در این بخش به بررسی دقیق‌تر مدل SQ-PUF پیشنهادی می‌پردازیم، از جمله یکنواختی و یکتایی مدل پیشنهادی، تأثیر تعداد ماژول SQ بر مقاومت مدل، مقاومت مدل در برابر حمله برحسب تعداد داده‌ها، مقایسه مقاومت مدل با سایر روش‌ها و سربار سخت‌افزاری مدل مورد بررسی دقیق‌تر قرار می‌گیرد.

۵-۴-۱- آنالیز یکنواختی[1] در SQ-PUF

یکنواختی عبارت است از تناسب وجود صفرها و یک‌ها در پاسخ‌های حاصل از PUF موردنظر که ایدئال این معیار برابر ۵۰٪ می‌باشد. یکنواختی یکی از مهم‌ترین معیارها برای طراحی یک مدل جدید PUF است که تعیین می‌کند که آیا صفرها و یک‌ها بین پاسخ‌ها به‌صورت یکنواخت توزیع‌شده است یا خیر.

ما برای محاسبه این مقدار برای مدل اصلی Arbiter PUF و مدل ارائه‌شده SQ-PUF از تعداد ۲×۱۰^{۴} چالش-پاسخ مختلف استفاده کردیم، همچنین درصد گزارش‌شده در جدول (۵-۲) بر اساس تعداد یک‌ها نسبت به‌کل پاسخ‌ها می‌باشد. همان‌طور که از گزارش نیز مشخص است ماژول SQ تأثیر خاصی روی یکنواختی مقادیر نمی‌گذارد.

[1] Uniformity

۵-۴-۲- آنالیز یکتایی[1] در SQ-PUF

یکتایی عبارت است از ارزیابی تفاوت پاسخ‌های تولیدشده توسط PUF های مختلف با اعمال یک چالش مشابه، یکی دیگر از معیارهای مهم در طراحی PUF ها است، همان‌طور که اثرانگشت هیچ دو فردی نمی‌تواند یکسان باشد، پاسخ مدل‌های مختلف PUF هم نباید به یک چالش یکسان باشد تا بتوان بر پایه آن احرازهویت را به‌درستی انجام داد، مقدار ایدئال برای این معیار برابر ۵۰٪ می‌باشد، با توجه به این‌که در SQ-PUF مقادیر وابسته به ترتیب هستند برای ارزیابی این معیار در هر مدل یکسان بودن ترتیب در تمامی حالت‌ها رعایت شده است.

از طرف دیگر با توجه به این‌که در مدل ارائه‌شده برای هر چالش دو پاسخ صحیح متفاوت وجود دارد احتمال وجود پاسخ‌های مشابه چهار برابر افزایش میابد. می‌توان این احتمال را به‌صورت $۴ \times p^n$ بیان کرد که n تعداد بیت پاسخ و p احتمال تشابه یک پاسخ در دو PUF مختلف می‌باشد. حال اگر p را برابر ۵۰٪ و n را برابر ۶۴ در نظر بگیریم حدوداً احتمال برابر شدن پاسخ‌ها در دو مدل PUF مختلف حدوداً برابر $۱۰^{-۱۹}$٪ می‌باشد.

در ادامه در جدول (۵-۲) مقادیر پارامترهای یکتایی و یکنواختی برای مدل اصلی PUF و مدل ارائه‌شده توسط ما و مدل R-PUF برای مقایسه ارائه‌شده است.

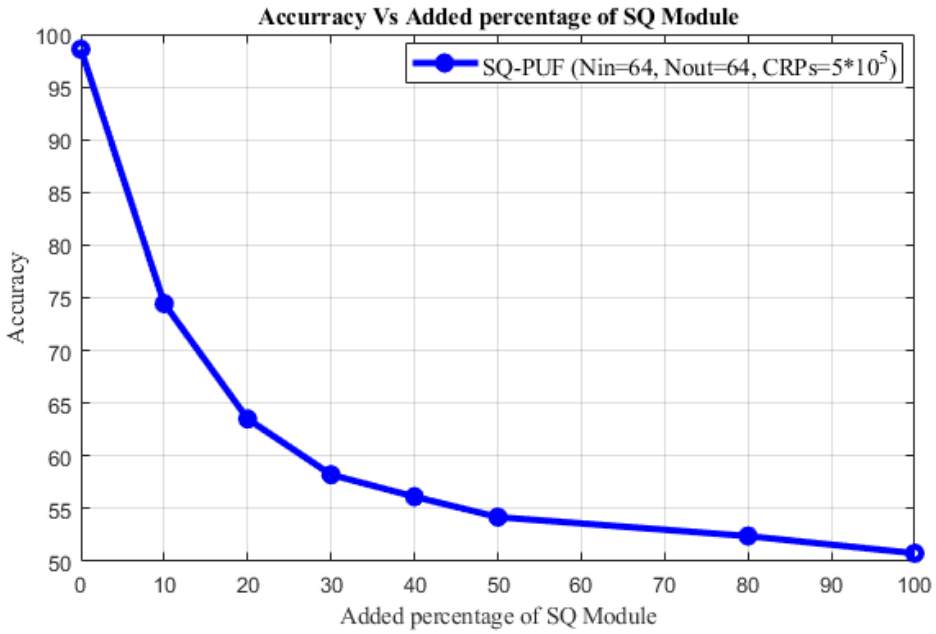
[1] Uniqueness

جدول (۵-۲) بررسی یکنواختی و یکتایی در مدل‌های مختلف PUF

	تعداد بیت‌های چالش	تعداد بیت‌های پاسخ	تعداد CRP-ها	یکنواختی (٪)	یکتایی (٪)
Arbiter PUF	64	64	۲×۱۰^{٤}	50.547%	49.197%
RPUF	64	64	۱×۱۰^{٤}	51.1%	49.7%
SQ-PUF	64	64	۲×۱۰^{٤}	50.561%	49.101%

۵-۴-۳- بررسی میزان تأثیر تعداد ماژول SQ در مقابله با حملات

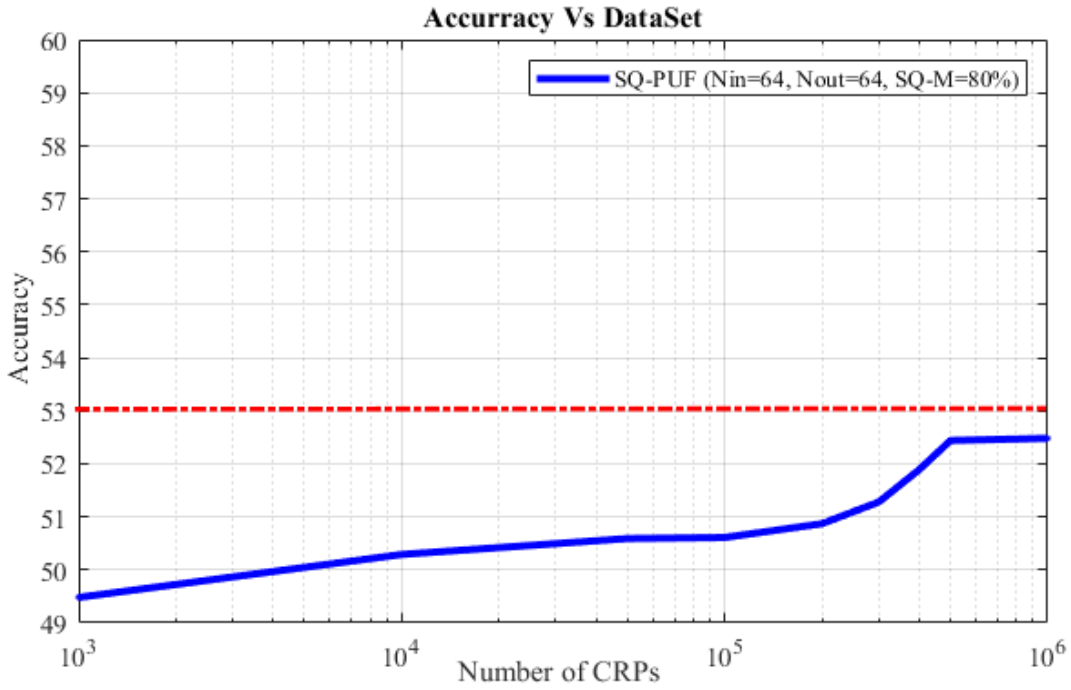
نمودار ارائه شده در شکل (۵-۳) تأثیر میزان استفاده از ماژول SQ در بیت‌های چالش را بر مقاومت مدل در برابر حملات یادگیری ماشین بیان می‌کند. این حملات توسط مدل CNN که قبلاً بیان شد و به‌وسیله دیتاستی به‌اندازه ۵×۱۰^{۵} انجام‌شده است. با این حجم از داده می‌توان گفت مقاومت کامل این مدل موردبررسی قرارگرفته است. همان‌طور که پیداست با اعمال ۸۰٪ ماژول SQ مدل PUF به مقاومت خوبی می‌رسد پس ما در ادامه شبیه‌سازی‌های خود را توسط مدل SQ-PUF با ۸۰٪ ماژول SQ انجام می‌دهیم.



شکل (۵-۳) نمودار تاثیر درصد اضافه کردن ماژول SQ بر مقاومت در برابر حملات

۵-۴-۴- بررسی مقاومت SQ-PUF

نمودار ارائه شده در شکل (۵-۴) میزان مقاومت مدل SQ-PUF با ۸۰٪ ماژول SQ نسبت به تعداد CRPs را نشان می‌دهد. این حملات توسط مدل CNN که قبلاً مطرح شد انجام‌گرفته است، همان‌طور که پیداست مدل ارائه‌شده با توجه به تعداد زیاد از CRP ها هم مقاومت خیلی خوبی از خود نشان داده است.



شکل (۵-۴) نمودار حمله به SQ-PUF با ۸۰٪ ماژول SQ و شبکه CNN پیشنهادی

۵-۴-۵- سربار سخت‌افزاری

جدول (۵-۳) سربار سخت‌افزاری بخش‌های مختلف یک مدل SQ-PUF برای ۶۴ بیت چالش که توسط FPGA پیاده‌سازی شده را نشان می‌دهد. این مدل شامل n ماژول SQ و یک RNG می‌باشد، ازآنجاکه معمولاً RNG در سیستم‌های الکترونیکی مورداستفاده است. می‌توان از RNG موجود در سیستم استفاده نمود. پس درواقع سربار سخت‌افزاری را می‌توان به‌صورت n*(LUT + FF) بیان کرد که n تعداد ماژول SQ استفاده‌شده در هر PUF است. همچنین برای مقایسه بهتر یک Basic-PUF نیز در گزارش آورده شده که نشان‌دهنده مزیت استفاده از LFSR در مقایسه با روش اصلی استفاده کمتر از پاسخ می‌باشد.

جدول (۵-۳) جدول گزارش سربار سخت‌افزاری

C (64) – R (64)	LUTs	FFs
Basic-PUF	8192	**64**
LFSR	34	**64**
PUF-64-1bit	128	**1**
SQ-Module	1	**1**

۵-۵- نتیجه‌گیری

جهت بررسی بهتر و مقایسه با سایر روش‌ها مدل SQ-PUF توسط روش‌های مختلف موردبررسی قرارگرفته که گزارش این بررسی در جدول (۵-۴) آمده است.

جدول (۵-۴) بیانگر مقاومت مدل SQ-PUF با ۸۰٪ ماژول SQ در برابر سه روش مختلف حملات پراستفاده در مقایسه با سایر مدل‌های PUF است. همان‌طور که پیداست مدل ما در برابر این‌گونه از حملات مقاوم می‌باشد همچنین لازم به ذکر است SQ-PUF توسط دیتاستی به‌اندازه 4×10^5 در تمامی روش‌ها موردحمله قرارگرفته است. همچنین برای بررسی بیشتر مدل ارائه‌شده با روش SVM هم موردبررسی قرار گرفت که دقت حمله در این روش هم از 50.22% فراتر نرفت.

جدول (۵-۴) جدول مقایسه مقاومت روش‌های مختلف در برابر حملات

	LR	ANN	CNN
Poly-PUF[38]	51.97%[38]	~95% [39]	-
OB-PUF[41]	71.99% [41]	~95% [39]	~96%
R-PUF(L2)[40]	~85% [39]	-	~83.3%
SQ-PUF	52.14%	52.4%	52.44%

طبق جدول (۵-۴) می‌توان نتیجه گرفت که مدل ارائه‌شده در این پایان‌نامه از مقاومت خوبی در برابر حملات یادگیری ماشین مطرح در این حوزه برخوردار است.

۵-۶- پیشنهادها

با توجه به اینکه بحث احرازهویت مبتنی بر PUF بحثی جدید و بسیار پراهمیت است، بنابراین در این حوزه می‌توان کارهای متعدد جدیدی ارائه نمود. همواره می‌توان با استفاده از آنالیز دقیق روش‌های مختلف نقاط ضعف آن‌ها را فهمید و در جهت رفع آن‌ها گام برداشت.

موارد زیر را می‌توان به عنوان کارهای آتی پیشنهاد داد:

- افزایش قابلیت اعتماد الگوریتم‌های احرازهویت با استفاده از ویژگی‌های دیگر
- انجام حملات مدل‌سازی مبتنی بر الگوریتم‌های یادگیری ماشین با استراتژی جدید بر روی مدل‌های مقاوم PUF
- استفاده از تکنیک‌های بهینه یادگیری ماشین برای تشخیص حملات احراز هویتی

فصل ۶: مراجع

[1] Y. Lu and L. Da Xu, “Internet of things (IoT) cybersecurity research: A review of current research topics,” *IEEE Internet Things J.*, vol. *6*, no. *2*, pp. *2103–2115*, Apr. *2019*.

[2] H. Gu and M. Potkonjak, “Efficient and Secure Group Key Management in IoT using Multistage Interconnected PUF,” *Proc. Int. Symp. Low Power Electron. Des.*, *2018*.

[3] P. Rughoobur and L. Nagowah, “A lightweight replay attack detection framework for battery depended IoT devices designed for healthcare,” in *2017 International Conference on Infocom Technologies and Unmanned Systems: Trends and Future Directions, ICTUS 2017*, *2018*, vol. *2018*-January, pp. *811-817*.

[4] G. E. Suh and S. Devadas, “Physical Unclonable Functions for Device Authentication and Secret Key Generation,” in *2007 44th ACM/IEEE Design Automation Conference*, *2007*, pp. *9-14*.

[5] B. Halak, M. Zwolinski, and M. S. Mispan, “Overview of PUF-Based hardware security solutions for the internet of things,” *Midwest Symp. Circuits Syst.*, vol. 0, no. May *2018*, *2016*.

[6] “*2020*’s Internet of Things Statistics, Facts & Predictions.” [Online]. Available: https://review*42*.com/resources/internet-of-things-stats/. [Accessed: *04*-Sep-*2021*].

[7] Mike Rosen, “Driving the Digital Agenda Requires Strategic Architecture.”

[8] “Future of IoT: *6* trends and predictions to watch.” [Online]. Available:

https://internetofthingsagenda.techtarget.com/feature/Future-of-IoT-6-trends-and-predictions-to-watch. [Accessed: *04*-Sep-*2021*].

[9] J. Y. Lee and J. Lee, “Current Research Trends in IoT Security: A Systematic Mapping Study,” *Mob. Inf. Syst.*, vol. *2021*.

[10] A. Government and D. of Home Affairs, “Code of Practice, Securing the Internet of Things for Consumers.” Australian government, *2020*.

[11] Alex Grant, “Securing the future of IoT,” *23*-Jun-*2021*. [Online]. Available: https://internetofthingsagenda.techtarget.com/post/Securing-the-future-of-IoT. [Accessed: *04*-Sep-*2021*].

[12] Knud Lasse Lueth, “State of the IoT *2020*: *12* billion IoT connections,” *2020*. [Online]. Available: https://iot-analytics.com/state-of-the-iot-*2020*-*12*-billion-iot-connections-surpassing-non-iot-for-the-first-time/. [Accessed: *04*-Sep-*2021*].

[13] R. Maes, *Physically unclonable functions: Constructions, properties and applications*, vol. *9783642413957*. Springer-Verlag Berlin Heidelberg, *2013*.

[14] M. El-Hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, “A taxonomy of PUF Schemes with a novel Arbiter-based PUF resisting machine learning attacks,” *Comput. Networks*, vol. *194*, no. May *2020*, p. *108133*, *2021*.

[15] Y. Cao, X. Zhao, W. Ye, Q. Han, and X. Pan, “A Compact and Low Power RO PUF with High Resilience to the EM Side-Channel Attack and the SVM Modelling Attack of Wireless Sensor Networks,” *Sensors 2018, Vol. 18, Page 322*, vol. *18*, no. *2*, p. *322*, Jan. *2018*.

[16] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, “Silicon physical random functions,” p. *148*, *2002*.

[17] A. Roelke and M. R. Stan, “Controlling the reliability of SRAM PUFs with directed NBTI aging and recovery,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. *26*, no. *10*, pp. *2016–2026*, Oct. *2018*.

[18] A. Ardakani, S. B. Shokouhi, and A. Reyhani-Masoleh, “Improving performance of FPGA-based SR-latch PUF using Transient Effect Ring Oscillator and programmable delay lines,” *Integration*, vol. *62*, pp. *371–381*, Jun. *2018*.

[19] J. Capovilla, M. Cortes, and G. Araujo, “Improving the statistical variability of delay-based physical unclonable functions,” *Proceedings, SBCCI 2015 - 28th Symp. Integr. Circuits Syst. Des. Chip Bahia*, Aug. *2015*.

[20] H. Zhang *et al.*, “Evaluation on flip-flop physical unclonable functions in a *14*/*16*-nm bulk FinFET technology,” *IEEE Int. Reliab. Phys. Symp. Proc.*, vol. *2018*-March, p. PPR. *11*-PPR. *15*, May *2018*.

[21] I. Papakonstantinou and N. Sklavos, “Physical Unclonable Functions (PUFs) design technologies: Advantages and trade offs,” *Comput. Netw. Secur. Essentials*, pp. *427–442*, Aug. *2017*.

[22] Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, and U. Rührmair, “The bistable ring PUF: A new architecture for strong physical unclonable functions,” *2011 IEEE Int. Symp. Hardware-Oriented Secur. Trust. HOST 2011*, pp. *134–141*, *2011*.

[23] M. Conti, A. Dehghantanha, K. Franke, and S. Watson, “Internet of Things security and forensics: Challenges and opportunities,” *Futur. Gener. Comput. Syst.*, vol. *78*, pp. *544–546*, Jan. *2018*.

[24] S. Balakrishnan, P. Wang, A. Bhuyan, and Z. Sun, “On success probability of eavesdropping attack in *802*. *11*ad mmWave WLAN,” *IEEE Int. Conf. Commun.*, vol. *2018*-May, Jul. *2018*.

[25] D. Lim, “Extracting Secret Keys from Integrated Circuits,” MIT, *2004*.

[26] D. Merli, D. Schuster, F. Stumpf, and G. Sigl, “Side-Channel Analysis of PUFs and Fuzzy Extractors,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. *6740*LNCS, pp. *33–47*, *2011*.

[27] L. Kusters and F. M. J. Willems, “Secret-key capacity regions for multiple enrollments with an SRAM-PUF,” *IEEE Trans. Inf. Forensics Secur.*, vol. *14*, no. *9*, pp. *2276–2287*, *2019*.

[28] K. A. Asha, A. Patyal, and H.-M. Chen, “Generation of PUF-Keys on FPGAs by K-means Frequency Clustering,” pp. *44–49*, Jan. *2019*.

[29] M. Huang, B. Yu, and S. Li, “PUF-Assisted Group Key Distribution Scheme for Software-Defined Wireless Sensor Networks,” *IEEE Commun. Lett.*, vol. *22*, no. *2*, pp. *404–407*, Feb. *2018*.

[30] M. D. M. Yu, R. Sowell, A. Singh, D. M’Raihi, and S. Devadas, “Performance metrics and empirical results of a PUF cryptographic key generation ASIC,” *Proc. 2012 IEEE Int. Symp. Hardware-Oriented Secur. Trust. HOST 2012*, pp. *108–115*, *2012*.

[31] T. Machida, D. Yamamoto, M. Iwamoto, and K. Sakiyama, “A New Arbiter PUF for Enhancing Unpredictability on FPGA,” *Sci. World J.*, vol. *2015*, *2015*.

[32] B. Chatterjee, D. Das, and S. Sen, “RF-PUF: IoT security enhancement through authentication of wireless nodes using in-situ machine learning,” in *Proceedings of the 2018 IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2018*, *2018*, pp. *205–208*.

[33] B. Chatterjee, D. Das, S. Maity, and S. Sen, “RF-PUF: Enhancing IoT Security Through Authentication of Wireless Nodes Using In-Situ Machine Learning,” *IEEE Internet Things J.*, vol. *6*, no. *1*, pp. *388–398*, Feb. *2019*.

[34] A. Ashtari, A. Shabani, and B. Alizadeh, “A new RF-PUF based authentication of internet of things using random forest classification,” *Proc. 16th Int. ISC Conf. Inf. Secur. Cryptology, Isc. 2019*, pp. *21–26*, Aug. *2019*.

[35] J. Delvaux, I. Verbauwhede, and D. Gu, “Security Analysis of PUF-Based Key Generation and Entity Authentication,” *2017*.

[36] M. Majzoobi, M. Rostami, F. Koushanfar, D. S. Wallach, and S. Devadas, “Slender PUF protocol: A lightweight, robust, and secure authentication by substring matching,” *Proc. - IEEE CS Secur. Priv. Work. SPW 2012*, pp. *33–44*, *2012*.

[37] G. T. Becker and R. Kumar, “Active and Passive Side-Channel Attacks on Delay Based PUF Designs.,” *IACR Cryptol. ePrint Arch.*, vol. *2014*, p. *287*, *2014*.

[38] S. T. C. Konigsmark, D. Chen, and M. D. F. Wong, “PolyPUF: Physically Secure Self-Divergence,” *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. *35*, no. *7*, pp. *1053–1066*, Jul. *2016*.

[39] J. Delvaux, “Machine-Learning Attacks on PolyPUFs, OB-PUFs, RPUFs, LHS-PUFs, and PUF-FSMs,” *IEEE Trans. Inf. Forensics Secur.*, vol. *14*, no. *8*, pp. *2043–2058*, Aug. *2019*.

[40] J. Ye, Y. Hu, and X. Li, “RPUF: Physical unclonable function with randomized challenge to resist modeling attack,” in *Proceedings of the 2016 IEEE Asian Hardware Oriented Security and Trust Symposium, AsianHOST 2016*, *2017*.

[41] Y. Gao *et al.*, “Obfuscated challenge-response: A secure lightweight authentication mechanism for PUF-based pervasive devices,” in *2016 IEEE International Conference on Pervasive Computing and Communication*

Workshops, PerCom Workshops 2016.

[42] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in *Proceedings of the ACM Conference on Computer and Communications Security*, *2010*, pp. *237–249*.

[43] Z. Wu, H. Patel, M. Tripunitara, and M. Sachdev, "Modeling Attacks on State-of-the-art PUF Architectures," *2018*.

[44] U. Rührmair and J. Sölter, "PUF modeling attacks: An introduction and overview," in *Proceedings -Design, Automation and Test in Europe, DATE, 2014*.

[45] Y. Wen and Y. Lao, "PUF Modeling Attack using Active Learning," in *Proceedings - IEEE International Symposium on Circuits and Systems*, *2018*, vol. *2018*-May.

[46] S. S. Zalivaka, A. A. Ivaniuk, and C. H. Chang, "FPGA implementation of modeling attack resistant arbiter PUF with enhanced reliability," in *Proceedings - International Symposium on Quality Electronic Design, ISQED, 2017*, pp. *313–318*.

[47] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Lightweight secure PUFs," in *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD, 2008*, pp. *670–673*.

[48] I. G. Târşa, G. D. Budariu, and C. Grozea, "Study on a true random number generator design for FPGA," *2010 8th Int. Conf. Commun. COMM 2010*, pp. *461–464*, *2010*.

[49] T. Arciuolo and K. M. Elleithy, "Parallel, True Random Number Generator (P-TRNG): Using Parallelism for Fast True Random Number Generation in Hardware," *2021 IEEE 11th Annu. Comput. Commun. Work. Conf. CCWC 2021*, pp. *987–992*, Jan. *2021*.

[50] B. Perach and S. Kvatinsky, "An Asynchronous and Low-Power True Random Number Generator using STT-MTJ," pp. *1–1*, Sep. *2020*.

[51] B. Acar and S. Ergün, "A random number generator based on irregular sampling and transient effect ring oscillators," *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. *2020*-October, *2020*.

[52] A. F. M. Razy, S. Z. M. Naziri, R. C. Ismail, and N. Idris, "Investigation and

design of the efficient hardware-based RNG for cryptographic applications," *2014 2nd Int. Conf. Electron. Des. ICED 2014*, pp. *255–260*, Jan. *2011*.

[53] R. S. Durga, C. K. Rashmika, O. N. V. Madhumitha, D. G. Suvetha, B. Tanmai, and N. Mohankumar, "Design and synthesis of LFSR based random number generator," *Proc. 3rd Int. Conf. Smart Syst. Inven. Technol. ICSSIT 2020*, pp. *438–442*, Aug. *2020*.

Abstract

The IoT industry faces many challenges, one of the most important of which is communication security. Also, one of the most critical actions for establishing security in communications is authentication; This means who sent the received message. Due to replay attack and other attacks in this area, the attacker can send the incomprehensible message received at the right time and achieve his sinister goals even if he achieves an encrypted message. Therefore, even the complexity of data encryption does not lead to network security. As a result, the use of cryptographic schemes alone can not be a good solution for authentication. The proposed method for authentication is to use the physical parameters. These parameters are randomly generated in the manufacturing process, so it is uncontrollable to make changes. These protocols are called Physical Unclonable Function. In this study, the resistance of some of these protocols has been investigated. Due to the advantages and disadvantages of these models, a new model called SQ-PUF has been introduced, which can show good resistance to attacks based on machine learning. This is a secure and scalable architecture.

Finally, we examined the resistance of this model against machine learning attacks and reports on the uniformity and uniqueness of data and hardware overhead of the above model.

Keywords: **IoT, network security, hardware security, authentication, machine learning**.

University of Tehran

College of Engineering

Faculty of Electrical and Computer engineering

Implementation of Machine-Learning-based Attacks on Physically Unclonable Functions (PUF)

By:

Abolfazl Sajadi

Supervisor:

Dr. Bijan Alizadeh

A thesis submitted to the Graduate Studies Office
In partial fulfillment of the requirements for The degree
of Master of Electronic Digital Systems in Electrical Engineering

September 2021