تحلیل مقاله در رابطه با درس معماری پیشرفته

تهیه و ارائه از: مسعود باقری

دوره کارشناسی ارشد

CARE: Lightweight Attack Resilient Secure Boot Architecture with Onboard Recovery for RISC-V based SOC

Code Authentication and Resilience Engine

CARE:یک معماری بوت امن سبک مقاوم در برابر حمله با بازیابی یکپارچه برایSOCمبتنی بر RISC-V

دیو آوانی

تعریف مسأله و هدف اصلی مقاله:

سوال اصلی مطرح شده در مقاله:

- با توجه به رشد سرسام آور اینترنت اشیا (IoT) و استفاده گسترده از سیستم‌های کامپیوتر توکار و سیستم‌های روی تراشه، بحث قابلیت اتکا برای این اشیاء هوشمند تبدیل به یک چالش اصلی و همیشگی شده.
- اتکا پذیری ویژگی یک سیستم کامپیوتری است که جنبه های مختلفی از جمله: ایمنی، امنیت، محرمانگی، دسترس‌پذیری، قابلیت اطمینان و نگهداری را شامل می شود. تحمل‌پذیری خطا مکانیزمی است که یک سیستم را قابل اتکا می‌نماید.
- اگر قبل از بالا آمدن یا روشن شدن این سیستم‌ها، خطایی رخ داده باشد و یا تهاجم و تخریبی صورت گرفته باشد آیا می توان این اشیا را آنقدر هوشمند ساخت که قابلیت خودبازیابی بدون دخالت انسان را داشته باشند؟

چه مشکلی باید برطرف شود؟

- پیشرفت های اخیر فن آوری، باعث افزایش فاجعه بار استفاده از دستگاه های کوچک تعبیه شده و IoT، در کاربردهای مختلفی از سیستم های کنترل صنعتی، سنجش و تحریک توزیع شده، سیستم های اتوماسیون خانگی و وسایل نقلیه، شده است.
- این افزایش استفاده و ارتباط متقابل [این دستگاهها] برای جمع آوری، انتقال و پردازش اطلاعات مهم امنیتی، دستگاه های کوچک تعبیه شده و اینترنت اشیا را به اهدافی جذاب برای حملات هکرها تبدیل کرده است.
- از نمونه های برجسته این حملات می توان به روتکیت [۱]، حملات بوت امن و بایوس [۲]، استاکس نت [۳] و هک جیپ [۴] اشاره کرد.
- دستگاه قربانی معمولا به سبب از کار افتادن ابزارهای تعمیر و بازیابی از راه دور، می‌بایست از شبکه جدا شده و به صورت دستی فلش شود، که وقت انرژی و خسارت زیادی ممکن است به دنبال داشته باشد.

چه ضرورتی برای مطرح شدن مسئله است؟

- چندین روش راه اندازی امن بر اساس سخت افزار، نرم افزار و طراحی مشترک سخت افزار/ نرم افزار ارائه شده است. در حالی که تکنیک های قبلی بوت امن بر روی شناسایی حملات مخرب تغییر کد تمرکز دارند، اما مسئله پاکسازی دستگاه های آسیب دیده کاملاً نادیده گرفته شده است.
- یک دستگاه آسیب دیده، برای بازیابی حالت عملیاتی خود به فلش مجدد کد به صورت دستی یا وایرلس (از راه دور) نیاز دارد.
- یک مهاجم هوشمند می‌تواند با خراب کردن پشته شبکه، از فلش مجدد از راه دور کد جلوگیری کند. این کار نیاز به مداخله دستی دارد.
- بعضی اوقات به دلیل قرارگیری دستگاهها (در سنسورها و دوربین های امنیتی منزل، سیستم های کنترل صنعتی و خودرویی، کشتی ها) ، فلش مجدد دستی نسبتاً دشوار می شود.
- گذشته از دشواری، به علت تأخیر حاصل از بازیابی دستی، می تواند وقت کافی برای مقاصد شوم مهاجم را در اختیارش قرار دهد.

چه روشهایی قبلا برای این کار انجام شده؟ (۱)

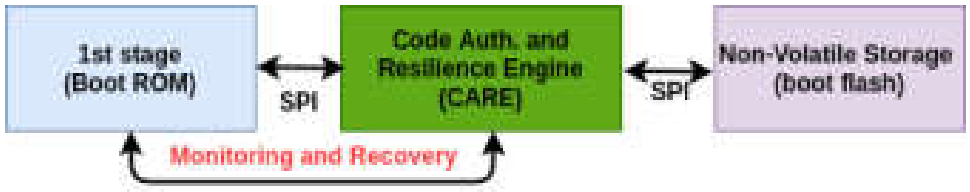
- اکثر اجراهای قبلی سیستم‌های بوت امن یکی از دو روش بوت اندازه گیری شده، یا احراز هویت شده را انجام می‌دهند و تعداد کمی از آنها هر دو را انجام می‌دهند.
- آرباغ و همکاران، اولین مکانیزم راه اندازی ایمن را پیشنهاد کرده است که یکپارچگی سیستم را با تأیید یکپارچگی کد نرم افزار بوت یا راه انداز، به صورت چند مرحله‌ای اندازه گیری می‌کند.
- رابط فریمور قابل توسعه یکپارچه UEFI از نسخه ۲.۲ بوت امن را به عنوان فرایندی برای بررسی یکپارچگی و اصالت هر مرحله از فرآیند بوت با محاسبه دایجست (digest) و مقایسه نتیجه با یک امضای رمزنگاری، تعریف می کند.
- یکی از روش‌های محبوب برای بوت امن، استفاده از یک پردازنده مشترک مجزا به نام TPM است. TPM دارای ثبات‌های ویژه و خاص منظوره‌ای به نام PCR است که نمی تواند بازنویسی یا جایگزین شود. PCR فقط با هش کردن اندازه گیری‌های نرم افزار همراه با مقادیر قبلی PCR قابل توسعه یا تغییر است. Platform Configuration Records
- پردازنده اینتل، از دو حالت بوت امن اندازه گیری شده و تأیید شده پشتیبانی می‌کند و از میکرو کد به عنوان ریشه اعتماد RoT استفاده می کند. برای بوت اندازه گیری شده ، از TPM استفاده می کند و برای بوت تأیید شده، هر جزء، توسط کلید سازنده امضا می‌شود و قبل از بارگذاری آن جزء، امضاها تأیید می‌شوند.

چه روشهایی قبلا برای این کار انجام شده؟ (۲)

- Microsoft’s fTPM [۱۰] یک استفاده موردی و موقعیت خاص از منطقه امن Arm مبتنی بر بوت و تصدیق امن را فراهم می کند.
- RISC-V مبتنی بر Sanctum [۷] از بوت امن و تصدیق از راه دور مبتنی بر نرم افزار استفاده می‌کند.
- SMART [۸] ، معماری RoT دینامیکی را برای دستگاههای سطح پایین فراهم می‌کند.
- کی استون Keystone [۱۲] یک استفاده موردی از محیط اجرایی قابل اعتماد TEE را با انکلاوها [حافظه های محصور یا پنهان شده] به نمایش می گذارد.
- حاج و همکاران [۵] معماری بوت امن مبتنی بر سخت افزار را برای SoC مبتنی برRISC-V ارائه می‌دهد.
- پروژه متن باز ریشه اعتماد اخیر گوگل [۱۱] نمونه ای از پیاده سازی RoT امن را ارائه می دهد.
- نویسنده ادعا دارد که، کار پیشنهادی، اولین اجرای یک معماری بوت امن سبک با موتور انعطاف پذیر یکپارچه و بازیابی توکار برای دستگاه‌های کوچک تعبیه شده و اینترنت اشیا است.

روش پیشنهادی ارائه شده چیست؟

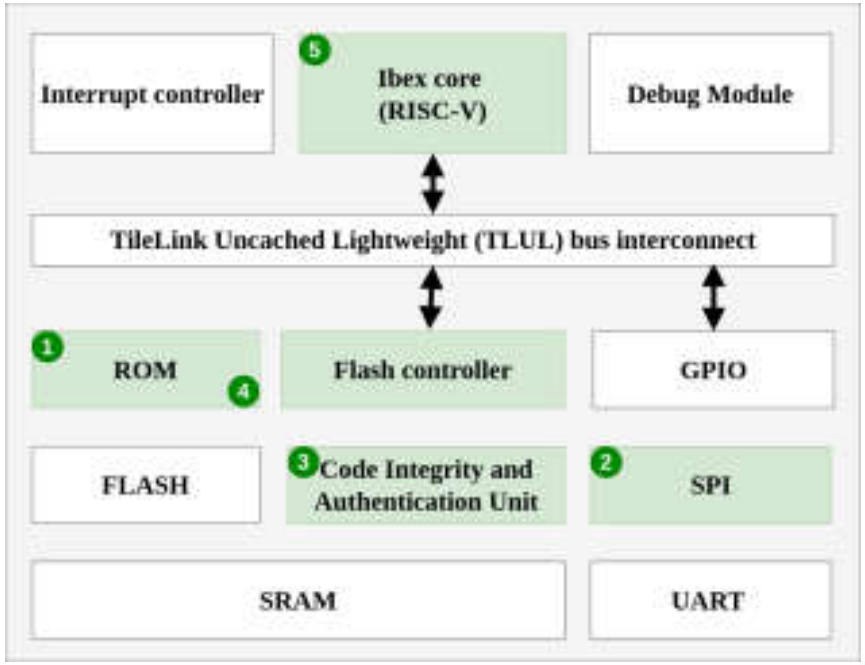
- ماژول ترکیبی CARE دارای دو جز اصلی
  - واحد احراز هویت یا اعتبارسنجی کد (CA)
  - موتور انعطاف پذیری (RE)
- این چارچوب با استفاده از زیر ماژول CA یکپارچگی و اصالت flash image را اندازه گیری می‌کند
- با شناسایی وجود کد مخرب، زیر ماژول RE را به کار می‌اندازد
- در غیر این صورت روند راه‌اندازی بعدی را ادامه می‌دهد
- کار RE فقط شناسایی مناطق خراب شده حافظه فلش و فلش مجدد انحصاری آن مناطق با کد خوب شناخته شده از ROM امن (پشتیبان) می‌باشد
- این چارچوب با استفاده از سیاست‌های کنترل دسترسی با استفاده از مکانیزم محافظت از حافظه فیزیکی (PMP) پردازنده RISC-V، درصورت تشخیص خطا، اجرای کد و نوشتن غیر مجاز را از RAM غیرفعال می‌کند
- و روند راه‌اندازی امن را تا جایی که کلیه بلوک‌های کد، صحت سنجی و بارگذاری شوند، ادامه می‌دهد.



توضیح راه حل پیشنهادی مقاله برای حل مسئله:

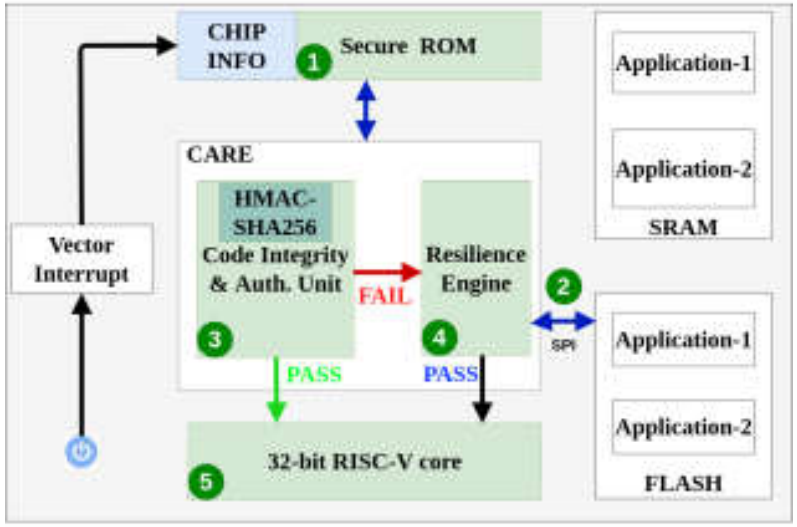
عملکرد سیستم به ۳ مرحله اصلی تقسیم می‌شود:

1. مقدار دهی اولیه سیستم
2. بررسی یکپارچگی و اصالت کد (CA)
3. موتور انعطاف پذیری (RE)

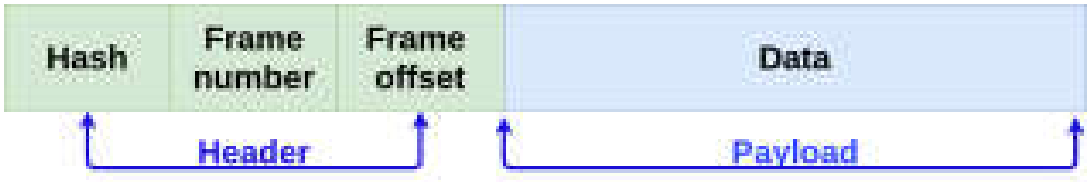


۱. مقدار دهی اولیه

- روشن شدن پاور
- کد FSBL را از ROM امن برای یافتن SPI و کنترل کننده‌های فلش پیدا و اجرا می‌کند.
- قوانین PMP را اعمال نموده و خواندن، نوشتن و اجرای کد غیر مجاز از حافظه محافظت نشده را مسدود می‌نماید.
- اطلاعات تراشه و کلید اشتراک متقارن را می‌خواند.
- کلیدهای مشتق شده تولید می‌کند.
- کنترل را به بوت استرپ منتقل می‌کند.



۲. بررسی یکپارچگی و اصالت کد (بوت استرپ)



بوت امن با فرآیند بوت استرپ شروع می‌شود و هربار که سیستم روشن یا بازنشانی شود، فرایند بوت استرپ اجرا می‌شود.

- تصویر فلش اجرایی به قطعات قاب داده ۱ کیلوبایتی تقسیم می‌شود و از طریق گذرگاه SPI به ترتیب به میزبان ارسال می‌شود.
- هر قاب داده از یک هدر و پی لود یا بسته اطلاعاتی مربوطه تشکیل شده است.
- هدر حاوی خلاصه امضا شده قاب داده‌ها است.
- قسمت مکان افست، محل حافظه فلش را نشان می‌دهد، که برای فلش مجدد کد استفاده می‌شود.
- پی لود یا بسته اطلاعاتی اصلی شامل ۹۶۸ بایت داده برای هر قاب است.

۲. بررسی یکپارچگی و اصالت کد (CA)

- فریمورک پیشنهادی از الگوریتم رمزنگاری HMAC-SHA256 بهره می‌برد.
- کد اصالت سنجی برپایه درهم سازی HMAC مانند هر MAC دیگری قادر است تا جامعیت داده و اعتبار پیام را همزمان بررسی کند.
- تابع درهم ساز SHA256 خلاصه هر قاب را محاسبه می‌کند.
- از یک ماژول سخت افزاری HW-SHA256 به عنوان هسته رمزنگاری استفاده شده.
- از یک کلید مشتق شده (بدست آمده از مرحله ۱) برای امضای خلاصه محاسبه شده استفاده می‌شود.
- امضای بدست آمده با هش موجود در فریم برای بررسی اصالت و یکپارچگی داده، مقایسه می‌شود.
- این فرایند برای تمام قاب‌های بعدی تکرار می‌شود.
- اگر همه چیز درست بود دستگاه با کد تأیید شده بوت می شود و در غیر اینصورت موتور انعطاف پذیری (RE) تحریک می‌شود.

۳. موتور انعطاف پذیری(RE)

1. شماره قاب و محل جابجایی قاب خراب را مشخص می‌کند و داده‌های قاب طلایی مربوطه را از EEPROMامن واکشی می‌کند.
2. منطقه منحصر به فرد حافظه فلش خراب را با کد خوب شناخته شده، مجدداً فلش می‌کند.
3. دسترسی خواندن و نوشتن غیر مجاز به حافظه را با استفاده از سازوکار PMPقفل می‌کند.

- . با این مراحل اطمینان حاصل می‌شود که دستگاه پیشنهادی بدون توجه به حملات تغییر حافظه، همیشه با کد خوب شناخته شده راه‌اندازی می‌شود.

روش پیاده سازی شده برای حل مسئله مقاله به چه صورت است؟

پیاده‌سازی بر روی fpga بوده و با اندازه‌گیری مولفه‌های مختلف المان‌های به کار رفته ازجمله مصرف انرژی و زمان تأخیر انتشار مربوطه، عملکرد هسته پردازشی تعیین شده است که همزمان با مقایسه با سایر ایده‌های مطرح بررسی می‌شود.

۱. مقایسه کارایی در هسته مبتنی بر نرم‌افزار و سخت‌افزار:

Parameters	Software	Hardware
Cycles (c)	**47033**	**2926**
Frequency (F) (MHz)	100	100
Block (b)	256	256
Throughput (T) (Mbps)	.54	8.74
Time (μsec)	470.33	29.26
Energy Consumption (E)	197.06	12.25
Energy Efficiency	**92.68**	**0.358**

۲. مقایسه هسته رمزنگاری پیشنهادی با هسته‌های مشابه موجود:

Work	*Area*	Freq.	Energy Con.	Device
This work	2591	100	.012	Artix-7
Opentitan [11]	2693	100	.022	Artix-7
He et al.[20]	7219	116.24	1.20	Arria II GX
He et al.[20]	10918	87	1.80	Arria II GX
Juliato et al. [21]	2347	138.10	.48	Stratix III
Juliato et al. [22]	4281	67	.285	CycloneII
Juliato et al. [22]	6874	41.25	.431	CycloneII

۳.مقایسه زمان و انرژی مصرفی با و بدون زیر ماژول موتور انعطاف پذیری:

Parameters	Without *CARE*	With *CARE*
Cycles req. for the first frame (c)	553611	576083
Cycles (rest of frames) (c)	103330	133790
Total Cycles (C)	**656941**	**709873**
Frequency (F) (MHz)	100	100
Time (t) (μsec)	**6569.41**	**7098.73**
Energy Consumption (E)	**2752.58**	**2974.36**

Time difference D_{Δ} = **529.32** μsec

۴. مقایسه کمی با پیشرفته‌ترین یا آخرین راه‌حل‌ها:

Parameters	*CARE*	Optitan[11]	Haj[5]	Ref.[23]
Design Type	Hybrid	Hybrid	HW	Hybrid
Secure boot function	hmacSha2	hmacSha2	Sha3	Sha2
Rom for Secure boot	yes	yes	yes	no/TPM
Integrity Check	yes	yes	yes	yes
Authenticity Check	yes	no	yes	yes
Recovery	yes	no	no	no
Lightweight	yes	yes	no	no

نقاط قوت و ضعف مقاله:

- توضیح مسأله و راه حل‌های موجود بسیار قوی بود و بطور گسترده‌ای تمام انواع تهدیدها و همچنین سبک‌های مختلف مقابله را دربرمی‌گرفت. جزئیات پیاده‌سازی سخت‌افزاری و نرم‌افزاری و در برخی موارد حتی کلیات آن نیز به هیچ وجه ارائه نشده. مثل کدهای زبان C، FPGAو ...

جمع بندی و پیشنهادات برای کارهای آتی:

- با توجه به استفاده گسترده از بردهایی همچون رسپبری‌پای، آردینو، اورنج‌پای، بیگل‌بن، تینکربورد و ... و با تجه به ساختار و عملکرد این بوردها شاید قدم بعدی در توسعه موضوع مقاله روشی باشد که به دستگاه اجازه دهد از روی کارت SDیا درایو فلش USBبوت شود.
- درضمن با توجه به محدودیت این روش در دریافت به‌روزرسانی، یکی دیگر از ایده‌ها می‌تواند اتخاذ ترتیبی برای حفظ امنیت دستگاه، درحین دریافت به‌روزرسانی باشد. به این ترتیب دامنه استفاده از این روش در دستگاه‌های بیشتری توسعه خواهد یافت.

خدا یار و نگهدار شما