اولین کنفرانس ملی ایده های نو در مهندسی برق
۱۶ و ۱۷ آذر ماه ۱۳۹۱- دانشگاه آزاد اسلامی واحد خوراسگان- اصفهان
NCNIEE
NATIONAL CONFERENCE NEW IDEA IN ELECTRICAL ENGINEERING

معماری یک کارت هوشمند کپی ناپذیر ایمن و قابل اعتماد مناسب برای سیستمهای بانکی

فرهاد فرخ پناه کلاش[۱]، رضا ابراهیمی آتانی[۲]
[۱]دانشگاه آزاد اراک، Farhad152000@yahoo.com
[۲]دانشگاه گیلان، rebrahimi@guilan.ac.ir

چکیده – در این مقاله ابتدا ساختار کلی کارت هوشمند بررسی شده و نقاط ضعف و قوت کارت دیده شده است. نقاط ضعف کارت هوشمند بانکی در ۲ بخش سخت افزاری و نرم افزاری می باشد، که جهت رفع آنها کارهایی صورت می پذیرد. در کارت هوشمند مهمترین کار برقراری ارتباط ایمن با کامپیوتر و سرور خارج از دسترس و محدوده آن می باشد. برای ارتباط برقرار کردن با محیط خارج، مهمترین رکن؛ داشتن کلیدی است که با آن بتوان عمل رمزنگاری و تصدیق انجام گیرد و به راحتی هم پیدا نشود. علاوه بر این حملات سخت افزاری که ممکن است به خود کارت وارد شود با استفاده از *PUF* جلوگیری می شود. برای بر قراری ارتباط با شبکه های خارج از *ATM* های موجود، واحد های بررسی درستی و رمزنگاری حافظه بکار می روند که با استفاده از کلید ایجاد شده می توانند رمزنگاری یا تصدیق نمایند و جداسازی شوند و با استفاده از چک کردن دسترسی در *MMU* از دسترسی به واحدها و پردازش های غیر مجاز جلوگیری می شود و از حملات نرم افزاری جلوگیری می شود. همه این حفاظت ها با اعمال معماری آن به قسمت *Co-processor* کارت هوشمند ایجاد می شود.

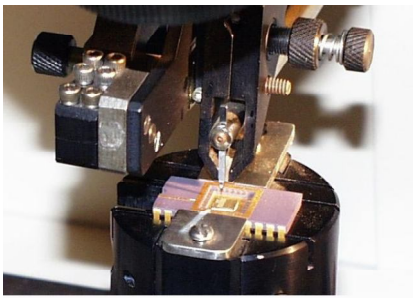
کلید واژه- کارت هوشمند، *PUF*، امنیت سخت افزاری، بانکداری الکترونیک، حملات سخت افزاری، رمزنگاری.

۱- مقدمه

در بانکداری نوین و همچنین در کشور عزیز ما و بعد از طرح تحول اقتصادی کارتهای اعتباری بانکی نقش بسیار مهمی را در زندگی افراد ایفا کرده اند. بطور مثال برای پرداخت یک دوره یارانه مبلغ بسیار زیادی نقدینگی بین بانکها جابجا می شود که اکثریت این جابجایی توسط کارتهای اعتباری بانکی انجام می شود. البته در بانکهای ایران و در بسیاری از کشورهای دنیا از کارت اعتباری مغناطیسی استفاده می شود که به دلیل حجم پایین نگهداری اطلاعات و همچنین ناامن بودن این کارتها، عملا استفاده از آنها مقرون به صرفه نیست. به همین دلیل بانکها باید بستر سازی کنند تا از کارتهای هوشمند بانکی استفاده کنند. این پردازنده بدلیل وجود پردازشگر مرکزی و همچنین کمک پردازنده امنیت و حجم بالایی از اطلاعات را پشتیبانی می کند. اما این بدان معنا نیست که این چیپ بطور کامل ایمن می باشد. همین مسئله باعث می شود دزدان در پی راهی برای هک کردن کارتهای هوشمند و دست یافتن به پولهای داخل آن باشند. ما سعی داریم در این مقاله با طرح کلی کارت هوشمند، نقاط ضعف آن را به نقاط قوت تبدیل کنیم. همانطور که در چکیده نیز گفته شد برای جلوگیری از حملات مخرب و غیر مخرب باید راهی پیدا کرد که فرد نتواند به اطلاعات داخلی کارت هوشمند دسترسی پیدا کند.

همانطور که در شکلهای ۱ و ۲ نشان داده شده است این حملات دسترسی مستقیم یا حداقل دسترسی از راه دور به کارت دارند. در تعریف این حملات می خوانیم که یا با بازکردن بسته داخلی یا مثلا با تزریق خطا در پی دستیابی به کلید رمز هستند. همانطور که در چکیده نیز گفته شد این کلیدهای رمز در واقع قسمت مهمی از اطلاعات را در بر دارند، چرا که رمزنگاری های داخلی و خارجی کارت هوشمند با استفاده از این کلید انجام می شود که اگر این کلید به دست مخالف بیفتد، او نیز می تواند کارت را رمز گشایی و به اطلاعات داخلی آن دسترسی داشته باشد. ما در این مقاله می خواهیم راه حلی نشان بدهیم تا علاوه بر اینکه دشمن نتواند حتی با حملات مخرب و غیر مخرب به کلید رمز دسترسی داشته باشد، تبادل اطلاعات در شبکه های بین بانکی نیز ایمن گردد.

شکل ۱.تراشه را در اسید نیتریک بالای ۹۸ درصد قرارداده و بسته آنرا باز می کنیم



شکل ۲.کارت هوشمند باز شده با استفاده از میکروسکوپهای صنعتی بررسی و دستکاری می شود.

کارت بانکی به عنوان وسیله تبادل پول چند وقتی است که جای پول نقد را در بین مردم گرفته است. در این مقاله سعی داریم با نشان دادن نقاط ضعف کارت هوشمند؛ در ادامه بحث با استفاده از معماری های مختلف در صدد تقویت آن برآییم و کارت را از هر نوع حمله ایمن سازیم.

۲- کارت هوشمند

در این بخش کارتهای هوشمند بانکی را که حاوی پردازشگر مرکزی به منظور پردازش اطلاعات و کمک پردازنده به منظور حفاظت می باشد را؛ توضیح می دهیم. کارتهای ریزپردازنده قادر به فراهم کردن تابعی برای خواندن / نوشتن و بالا بردن امنیت در داخل کارت حاوی CPU می باشد. سبک معماری داخلی کارتهای ریزپردازنده بصورت قابل توجهی به کامپیوترهای شخصی شباهت دارد، اجزاء بلوک ساختمانی در این کارتها عبارتند از : CPU,ROM,RAM,I/O GATEWAY,E^2PROM در زیر به بررسی و توضیح آنها پرداخته می شود.

۱. CPU : اغلب در یک ریز پردازنده ۸بیتی همراه با باس آدرس ۱۶ بیتی بکار برده می شود. این ویژگی آنجا کارساز است که پردازنده کارت به آدرسهای بالایی تا حداکثر 64KB می تواند دسترسی داشته باشد.

۲. RAM :حافظه فراری است که احتیاج به توان برای حمایت از داده ها دارد. این کار معمولا برای ذخیره کردن در CPU می باشد. دلیل انتخاب سایز کوچک برای آن؛ این است که فضای بیشتری به E^2PROM,ROM اختصاص بگیرد؛ بطوریکه کل سایز آن به 25mm محدود شده است.

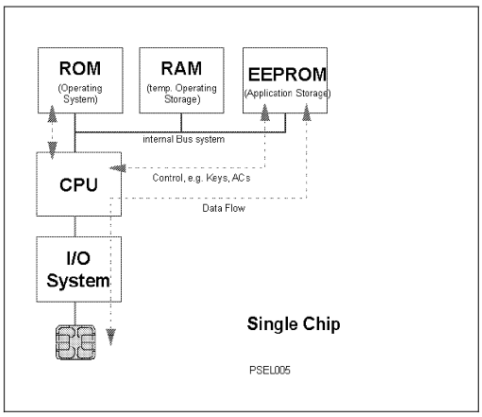
۳. ROM : حاوی سیستم عامل کارت هوشمند است و در هنگام تولید چیپ بارگذاری شده است. سایز آن از چند کیلو بایت تا 32kB متغیر است. پس از آنکه در سیستم عامل بار شد؛ امکان بهبودی[1] آن وجود ندارد.

۴. E^2PROM :حافظه غیر فرار است که برای نگهداری تمام داده ها و برنامه ها استفاده می شود. سایزی که بیشترین استفاده برای کاربران را دارد 8kB می باشد. کلید رمز در کارت هوشمند نیز در این حافظه ذخیره می شود.

۵. I/O GATEWAY : گذرگاه برای انتقال داده در یک روش سریال، بیت به بیت استفاده می شود.

در مورد کارتهای هوشمند تماسی باید نکاتی را بگوییم:

کارت باید در قرائتگر قرار داده شود تا اتصال فیزیکی با روکش طلایی برقرار شود و قرائتگر توان را به کارت بدهد و همچنین سیگنال کلاک به چیپ اعمال شود.

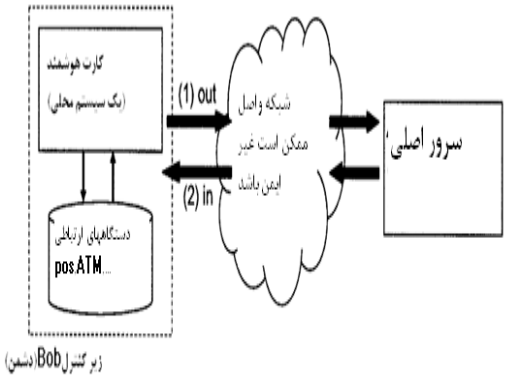


شکل۳. معماری داخلی کارت هوشمند

[1] Upgrade

۳- تعامل اطلاعات با شبکه های خارج از محدوده و مرز

همیشه و در همه جا(بالاخص در اینترنت) ایمنی و قابلیت اعتماد، درست بودن ارسال پیغام، درست بودن دریافت پیغام و فایل از اهمیت بالایی برخوردار می باشد. باید کاری انجام شود که حتی اگر ریزپردازنده زیر نظر یک دشمن باشد؛ پردازنده بتواند با کاربر خارج از محدوده ارتباط صحیح و درست برقرار کند.



شکل۴- چالش امنیتی اصلی حل شده بوسیله کارت هوشمند

در شکل ۴؛ ۳ مکان و قسمت مهم وجود دارد. سمت چپ که زیر کنترل Bob می باشد ساختار کلی ارتباط داخلی کارت با دستگاههایی نظیر POSوATMو ... می باشد؛ در وسط شکل یک شبکه واسط می باشد که وظیفه برقراری ارتباط بین ATM و سرور مرکزی را دارد، نا گفته نماند که ممکن است این شبکه به دلایل مختلفی موجب انشعاب اطلاعات شود و غیر قابل اعتماد شود. در انتهای سمت راست شکل، سرور مرکزی می باشد که محاسبات اصلی داخل آن انجام می شود و دستگاههای واسط با آن ارتباط برقرار می کنند، سرور احتیاج دارد که بداند اطلاعاتی که از سیستم محلی فرستاده می شود اطلاعاتی صحیح و بدون دستکاری می باشد(مرحله ۱) و همچنین اطلاعاتی که می فرستد بدون دخالت و کاملا صحیح به صورت محرمانه ذخیره شود(مرحله ۲).

۳-۱- بررسی

در اینجا ابتدا ساختار کلی کارت هوشمند نشان داده شد، در قسمت قبلی شبکه های تبادلی پولی را نشان دادیم؛ در ادامه انواع حملاتی که به کارتهای هوشمند موجود انجام می شود نشان داده می شود و بلافاصله معماری های مقاومتی در برابر آنها و اعمال شده به چیپ پردازنده نشان داده می شود

۴- تکنیکهای دستکاری در کارت هوشمند :

اصولا حملاتی که به منظور دستکاری روی کارت هوشمند انجام می شود، به ۲ نوع مخرب[2] و غیرمخرب[3] تقسیم می شوند

حملات مخرب:

همه تکنیکهای دستکاری و جستجو مخرب می باشند. این تکنیکها احتیاج به چندین ساعت یا در برخی موارد چندین هفته کار در یک آزمایشگاه مخصوص دارند که برای این کار حتما باید بسته اصلی آنها را خراب کرد. تکنیکی است که دسترسی به واسطهای چیپ بطور مستقیم دارد.

حملات غیرمخرب :

یک پردازنده یک مجموعه از چند صد فلیپ فلاپ است که حاوی وضعیت جاری آنها همراه با منطق ترکیبی است که از هر سیکلهای کلاک محاسبه می شود. برخی از تاثیرهای آنالوگ در این قبیل سیستمها در حملات غیر مخرب استفاده می شود. برخی از آنها عبارتند از:

حملات نرم افزاری: استفاده از واسطهای ارتباطی نرمال در پردازنده و استخراج آسیب های امنیتی پردازنده، الگوریتم های رمز نگاری، یا در پیاده سازی آنها در حمله به پردازنده کمک می کند.

استراق سمع: خواندن و مشاهده تکنیک، همراه با رزولاسیون بالا؛ شناسایی آنالوگ همه منابع و اتصالات داخلی و

[2] Invasive
[3] Non-Invasive

الکترومغناطیس های تولید شده دیگر بوسیله پردازنده در هنگام عملیات نرمال باعث حمله به پردازنده می شود.

تولید خطا: این تکنیک از محیط غیر عادی برای تولید خرابی در پردازنده استفاده می کند که محتوی دسترسی اضافه می باشد.

هر ترانزیستور و اتصالات داخلی یک توان و مقاومتی دارد که همراه با فاکتورهایی از قبیل درجه حرارت، ولتاژ منبع شناخته می شود که با استفاده از تاخیر بخش های سیگنال اندازه گیری می شوند. بنابر نوسان تولید پردازنده این مقدار می تواند در یک چیپ تکی و بین چیپ های مشابه تغییر یابد.

منبع جاری احتیاج دارد که توان بارگیری شده را یا شارژ یا خالی کند، هنگامیکه خروجی تغییر می یابد.

فلیپ فلاپ می تواند وضعیت جاری جدید را قبول کند؛ تنها بعد از اینکه خروجی منطق ترکیبی در وضعیت قبلی تثبیت شود.

اینها تاثیرهای مشهور دیگری هستند. در هنگام بازبینی امنیت طراحی پردازنده، اغلب کارهای جزییاتی وابسته به شبیه سازی آنالوگ و تست آنها کارهای مهمی در جلوگیری از حملات غیر مخرب می باشند. پردازنده کارت هوشمند اغلب به حملات غیر مخرب آسیب پذیر می باشد، چون مهاجم کنترل کامل به توان و منبع کلاک دارد. ماژولهای امنیتی بزرگتر می تواند همراه با باتری برای پشتیبانی کردن، پوشش شیلد و .... برای کاهش ریسکهای بسیاری که پردازنده کارت هوشمند را مورد هجوم قرار می دهند، استفاده شوند.

۵- حملات و ضد حملات

در این بخش می خواهیم بطور مفصل به حملاتی که به داخل یا خارج کارت هوشمند ممکن است انجام شود، بپردازیم و سپس به اقداماتی که می توان در برابر این حملات انجام داد، بپردازیم.

۵-۱- اقدام متقابل در برابر حملات مخرب و غیر مخرب

هدف اصلی و کلی حملات مخرب و غیر مخربی که می خواهند به داخل پردازنده دست یابند؛ دستیابی به کلید رمزی است که پردازنده با آن با پردازنده های دیگر یا سرور مرکزی تبادل اطلاعات می کند. در واقع مهمترین بخش هر کارت هوشمند کلید رمز آن است که بیشترین محافظت را باید بر روی آن انجام دهیم. در کارت هوشمند کلید رمز بطور غیر فرار و مستمر در E^2PROM ذخیره می شود و هنگامی که کارت به آن نیاز دارد؛ آنرا از حافظه بازیابی می کند. اما در حملات مخرب با برداشتن لایه رویی(همانطور که در مقدمه توضیح داده شد) و همچنین در حملات غیر مخرب با استفاده از حملاتی مثل تزریق خطا به کلید رمز دست پیدا می کنند و عملا کارت را کپی می کنند. ما از تابع کپی ناپذیر فیزیکی استفاده می کنیم که رمز را بطور فرار تولید کرده و در انتهای مرحله تولید رمز از بین میرود. اینکار باعث می شود زمان استفاده از آن کلید تا حد ممکن پایین بیاید و دشمن نتواند از آن استفاده کند.

۵-۱-۱- تابع تصادفی فیزیکی:

یک تابع غیر قابل کپی‌برداری فیزیکی تابعی است که چالشها را به پاسخها نگاشت می‌کند و به صورت یک ابزار فیزیکی مجسم می‌شود, که شامل خصوصیات زیر است:

ارزیابی آسان: ابزار فیزیکی می‌تواند تابع را در یک مدت کوتاه ارزیابی کند.

توصیف دشوار: به خاطر محدودیت تعداد اندازه‌گیریهای فیزیکی قابل قبول یا برخی اشکالات زوجهای چالش-پاسخ[۴] انتخاب شده، مهاجمی که از ابزار و منابع محدود(مثل زمان، پول، مادهٔ خام و . . .)استفاده می‌کند؛ فقط می‌تواند به مقدار قابل اغماضی از اطلاعات در مورد پاسخهای چالشها (که به صورت تصادفی انتخاب شده‌اند) دست یابد.

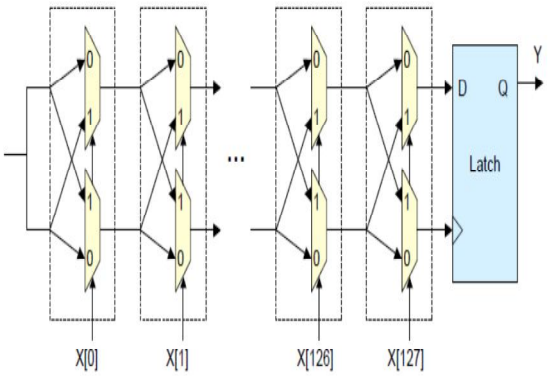
PUF فقط با سیستم فیزیکی ارزیابی می‌شود و برای هر نمونه فیزیکی، منحصر به فرد است. PUFها می‌توانند توسط سیستمهای فیزیکی مختلفی پیاده‌سازی شوند، مثلا PUFهای سیلیکونی[۵] مبتنی بر زمانبندی و تاخیر پنهانی اطلاعات مدارات مجتمع می‌باشند. حتی در صورت داشتن ماسکهای جانمایی یکسان، تفاوتها در حین پروسه ساخت باعث تغییرات تاخیر بر

[4] *Challenge-Response Pair*
[5] *Silicon PUFs*

روی تراشه‌های مختلف می‌شود

Arbiter PUF:

در شکل۵، مدار تاخیر یک PUF سیلیکونی که توسط یک سری مالتی‌پلکسر و یک arbiter(latch) ساخته شده، نشان داده شده است. مدار دارای چند بیت ورودی X می‌باشد و بیت خروجی Y بر اساس اختلاف تاخیر نسبی بین دو مسیر تعیین می‌شود. بیتهای ورودی، مسیر تاخیر را توسط مالتی‌پلکسرها کنترل می‌کنند. اگر بیت کنترلی X[i] صفر باشد، مالتی‌پلکسرها سیگنال را از بین دو مسیر حالت صفر مالتی‌پلکسر عبور می-دهند. در غیر این صورت مسیرهای بالا و پایین جابجا می‌شوند. بنابراین بر اساس هر مقدار X[i] دو مسیر تاخیر متفاوت داریم. به منظور ارزیابی خروجی برای یک ورودی خاص، یک لبه بالارونده سیگنال به طور همزمان به دو مسیر داده می‌شود و تاخیر دو مسیر به arbiter تعیین کننده سرعت سیگنال است. در صورتی که ورودی لچ (D) سریعتر باشد، خروجی ۱ و در غیر این صورت صفر می‌شود.



شکل۵- مدار تاخیر یک arbiter-PUF

علاوه بر این حفاظت می توان از حفاظت نرم افزاری داخل برنامه نیز استفاده کرد که برای جلوگیری از این ۲ نوع حمله؛ برنامه را بر اساس حفاظت؛ دسترسی و ... به مدهای مختلف تقسیم کرده و آنها را اجرا می کنیم. حملات مخرب که نیاز به باز کردن بسته و دسترسی به داخل پردازنده دارند از مد PTR و حملات غیر مخرب که دسترسی مستقیم به داخل پردازنده ندارند از مد TE در برنامه استفاده می کنند؛ مد TE هر مداخله ای که محتویات ناحیه را تغییر بدهد کشف می کند و مد PTR علاوه بر اینکه مداخله را کشف می کند؛ پوشیدگی را حفظ می کند. PTR بدان معناست که در شبکه هنگامی که در مبدا رمزنگاری شد در مقصد باید در سخت افزار ویژه و در نرم افزار ویژه رمزگشایی انجام شود.

۵-۲- حملات به کش های داخلی

کش های داخل تراشه می توانند به راحتی دستکاری شده؛ آدرسهای مجازی و فیزیکی آن جابجا شده و بر روی آنها تعدادی حمله انجام شود.

۵-۲-۱- اقدام متقابل:

کش های داخل تراشه بوسیله مکانیسم برچسب؛ محافظت داده می شود. هنگامیکه یک پردازنده دسترسی به یک بلوک ناحیه های بازبینی/ شخصی داشته باشد، بلوکها همراه با SPID پردازش علامت زده می شوند. پردازش های منظم بوسیله مقدار SPID که صفر است، نمایش داده می شوند. این SPID ویژه، مالکیت بلوک کش می باشد. هر بلوک کش همچنین حاوی آدرس مجازی متناظر می باشد که بوسیله مالک پردازنده در آخرین دسترسی به بلوک استفاده می شود. هنگامیکه یک پردازنده ایمن به یک بلوک کش؛ در کش دسترسی پیدا می کند که؛ محافظت درستی و تمامیت داده را احتیاج دارد(در ناحیه بازبینی)، پردازنده، برچسب بلوکها را قبل از آنکه از آن استفاده شود؛ چک می کند. اگر SPID فعال باشد، SPID را در بلوک کش منطبق می کند و اگر آدرس دسترسی مجازی با آدرس مجازی بلوک کش منطبق باشد، دسترسی ادامه پیدا می کند. بعبارت دیگر؛ مقدار بلوک کش بوسیله مکانیسم بررسی درستی داده خارج از تراشه بازبینی می شود. اگر درستی موفقیت آمیز باشد؛ SPID و آدرس مجازی بلوک به روز می شوند. حتی همراه با SPID ها و برچسبهای آدرس مجازی، سیستم عامل های مخرب می تواند یک حمله تکرار را بوسیله تغییر نگاشت مجازی به فیزیکی انجام بدهد، اگر کش ها به صورت فیزیکی آدرس دهی شده باشند.

۵-۳- حملات فیزیکی(سخت افزاری) و نرم افزاری

حافظه داخل/ خارج از تراشه:

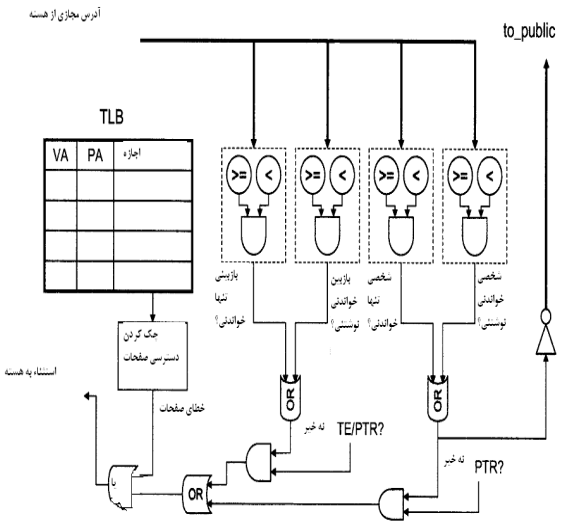
حملات سخت افزاری می تواند بصورت دلخواه محتویات حافظه خارج از تراشه را تغییر بدهند. همچنین، حملات نرم افزاری می تواند محتویات کش های داخل تراشه را تغییر بدهند. بنابراین، مهاجم می تواند دستورالعمل ها و داده ها را در حافظه؛

تکرار؛ جابجا و تعویض کند. بطور کلی؛ تمام فضای حافظه مجازی آسیب پذیر است.

۵-۳-۱- اقدام متقابل:

برای جلوگیری از حملات نرم افزاری ابتدا VM[6] فضای حافظه مجازی پردازشها را از یکدیگر جدا می کند که این باعث می شود برنامه ها در فضای مجازی یکدیگر نتوانند تداخل داشته باشند. همچنین با استفاده از هسته پردازنده زیر سیستم حافظه تغییر داده شده در نظر گرفته می شود. شکل ۶ واحد مدیریت حافظه(MMU) همراه با اجازه چک کردن دسترسی اضافی را شرح می دهد. MMU یک TLB قراردادی دارد که یک آدرس فیزیکی را به آدرس مجازی ترجمه می کند و اجازه دسترسی به صفحات را چک می کند. این جستجوی TLB یک سیکل در مرحله MEM در پردازنده پایپ لاین زمان می گیرد. در نتیجه، خطای صفحه در انتهای مرحله MEM کشف می شود

اگر چنین باشد، MMU بازبینی می کند که آیا دسترسی حافظه در مد اجرایی جاری اجازه داده می شود یا خیر.



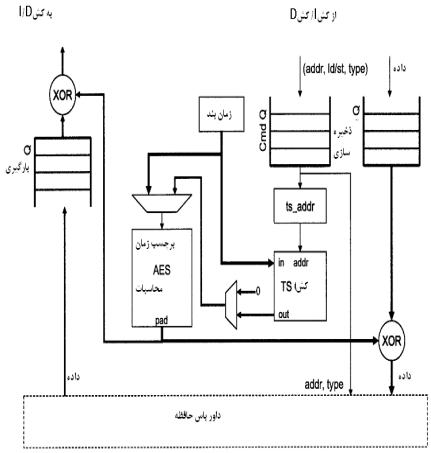
شکل۶- واحد مدیریت حافظه (MMU) همراه با الحاقیات امنیتی

۵-۳-۲- رمز نگاری:

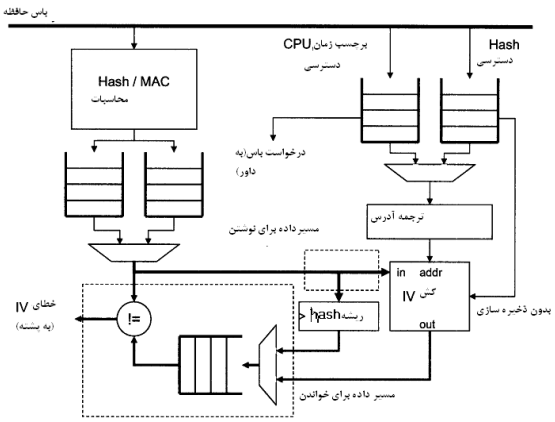
پیاده سازی واحد ME در یک فرم ساده شده نشان داده شده است. دسترسی حافظه از کش های دستورالعمل و داده در قسمت بالا و سمت راست واحد ME نشان داده شده است. واحد ME این دسترسی ها را در صف معمولی، بافر می کند و صف را ذخیره می کند. برای خواندن از کش I/D، واحد ME آدرس برچسب زمان متناظر را محاسبه می کند، برچسب زمان را از کش بر چسب زمان (TS) می خواند و با استفاده از AES رمز گشایی را انجام می دهد. در زمان مشابه، واحد ME یک درخواست را برای داده به باس حافظه می فرستد. هنگامیکه داده از حافظه وارد شد، در صف بارگیری بافر می شود(در قسمت چپ شکل) تا اینکه محاسبات AES کامل شود. هنگامیکه رمز گشایی آماده شد، داده بوسیله XOR کردن با دنباله[7] ؛رمزگشایی می شود و کش پردازنده بازگشت داده می شود. برای نوشتن از کش I/D، واحد ME از برچسب زمان رجیستر زمان بند آنها استفاده می کند و محاسبات رمزگشایی را انجام می دهد. در زمان مشابه، برچسب زمان جدید در کش برچسب زمان؛ ذخیره می شود. سپس رجیستر زمانبند یکی، یکی افزوده می شود. هنگامیکه دنباله رمزنگاری شده از واحد AES آماده شد، داده از پردازنده همراه با دنباله، رمز گشایی می شود و به حافظه فرستاده می شود. در حالیکه در این شکل نشان نداده ایم، واحد ME یک باز-رمزنگاری را انجام می دهد، هنگامیکه زمان بند به یک مقدار بیشینه نزدیک می شود. این باز رمز نگاری می تواند یا در نرم افزار یا در سخت افزار پیاده سازی شود. در پیاده سازی، واحد ME یک سیگنال را به هسته پردازنده می فرستد؛ هنگامیکه زمانبند به مقدار آستانه می رسد، که یک تله کارت هوشمند برای باز رمز نگاری اتفاق می افتد

[6] Virtual MEmory

[7] Pad



شکل۷- واحد رمزنگاری



شکل۸. واحد بررسی درستی داده

۵-۳-۳- بررسی درستی و تمامیت داده:

در نهایت، واحد IV در شکل پایین نشان داده شده است. عملیات اصلی در واحد IV نسبتا ساده است. واحد هر دسترسی به حافظه را در باس حافظه نشان می دهد. اگر یک دسترسی در حفاظت IV علامت زده شود (IV تنها خواندنی،IV خواندنی-نوشتنی، برچسبهای زمان یا hashها)، واحد IV، MAC یا چکیده ساز را در مقدار داده محاسبه می کند. در زمان مشابه، آدرسها و برچسبها بافر می شوند و آدرس فراداده[8] متناظر محاسبه می شود. برای یک دسترسی خواندن، واحد IV، hash ها یا MAC را یا از رجیستر چکیده ریشه یا کش IV می خواند. سپس این والد چکیده یا MAC با یک محاسبه از مقدار در باس حافظه مقایسه می گردد.

۶- نتیجه‌گیری

با استفاده از منابع بالا می توان درصد زیادی از حملات به کارت را خنثی کرد، به شرط آنکه دشمن دسترسی فیزیکی به پردازنده نداشته باشد. تمام معماریهای موجود را باید در قسمت کمک پردازنده کارت هوشمند قرار داد که با توجه به حجم منابع در این قسمت باید از مساحت و حجم قسمتهای دیگر کارت هوشمند کاست. با استفاده از bench marck می توان فهمید که برای قرار دادن پردازنده ایمن در کارت هوشمند به یک Coprocessor 1.5cm*1.5cm احتیاج داریم و تقریبا یک مقدار کمی گیت سربار احتیاج داریم که البته با توجه به حفاظت فوق العاده بالایی که بوسیله پردازنده ایجاد می شود سربار مطلوب می باشد.

مراجع

[1] Johan Agat. *Transforming out timing leaks*. In *27th ACM Principles of Programming Languages,* January 2000.
[2] Tiago Alves and Don Felton. *Trustzone: Integrated hardware and software security*. ARM white paper, July 2004.
[3] Ross Anderson. Security Engineering*: A Guide to Building Dependable Distributed Systems*. John Wiley and Sons, 2001.
[4] Ross Anderson and Markus Kuhn. *Tamper resistance - a cautionary note*. In *Proceedings* of the Second USENIX Workshop on Electronic Commerce, pages 1-11, November 1996.
[5] Ross Anderson and Markus Kuhn. *Low cost attacks on tamper resistant devices*. In IWSP: International Workshop on Security Protocols, LNCS, 1997.
[6] W. Arbaugh, D. Farber, and J. Smith. *A secure and reliable bootstrap architecture*. In Proceedings of the 1997 IEEE Symposium on Security and rivacy, pages 65-71, May 1997.
[7] T. Arnold and L. van Doorn. The IBM PCIXCC: *A new cryptographic co-processor*

[8] Meta data

for the IBM eServer. IBM Journal of Research and Development, 48:475-487, 2004.
[8] Divya Arora, Srivaths Ravi, Anand Raghunathan, and Niraj K. Jha. *Secure embeddedprocessing through hardware-assisted run-time monitoring*. In *DATE'05: Proceedings* of the conference on Design, Automation and Test in Europe, pages 178-183, Washington, DC, USA, 2005. IEEE Computer Society.
225
[9] Arash Baratloo, Timothy Tsai, and Navjot Singh. *Transparent run-time defense against stack smashing attack*s. In Proceedings of the USENIXAnnual Technical Conference, 2000.
[10] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. *Keying hash functions for message authentication.* In *CRYPTO '96,* volume 1109 of *LNCS.* Springer-Verlag, 1996.