دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)

بسم الله الرحمن الرحیم

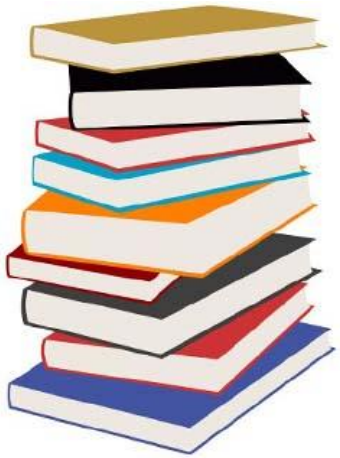
اِعمال پویای فوق‌خاصیت‌های امنیتی

سید محمدمهدی احمدپناه
smahmadpanah@aut.ac.ir

استاد راهنما: دکتر مهران سلیمان‌فلاح
استاد درس سمینار: دکتر بابک صادقیان

دانشگاه صنعتی امیرکبیر
۱۴ مهر ۱۳۹۵

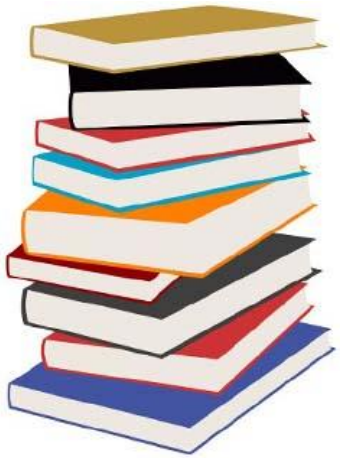
دانشکده مهندسی کامپیوتر
و فناوری اطلاعات



فهرست

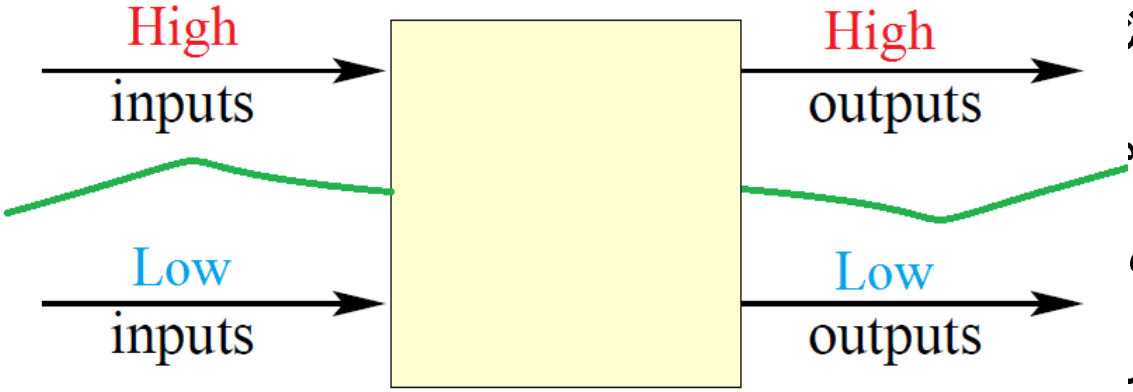
- مقدمه
- خط‌مشی امنیتی؛ خاصیت‌ها و فوق‌خاصیت‌ها
- دسته‌بندی کلی مکانیزم‌های اعمال خط‌مشی
- محورهای تأثیرگذار در توانایی ناظرها
- مدل‌سازی ناظرها به کمک خودکاره‌ها
- پارادایم‌های اعمال
- مقایسه توانایی ناظرها در محورها
- تأثیر محدودیت‌های محاسباتی و حافظه‌ای در توانایی ناظرها

فهرست



- تأثیر در اختیار داشتن اطلاعات ایستا بر توانایی ناظرها
- مقایسه تکنیک‌های اعمال زمان‌اجرا
- مقایسه مکانیزم‌های اعمال پویای خط‌مشی جریان اطلاعات
- جمع‌بندی
- مسائل باز و پروژه کارشناسی ارشد

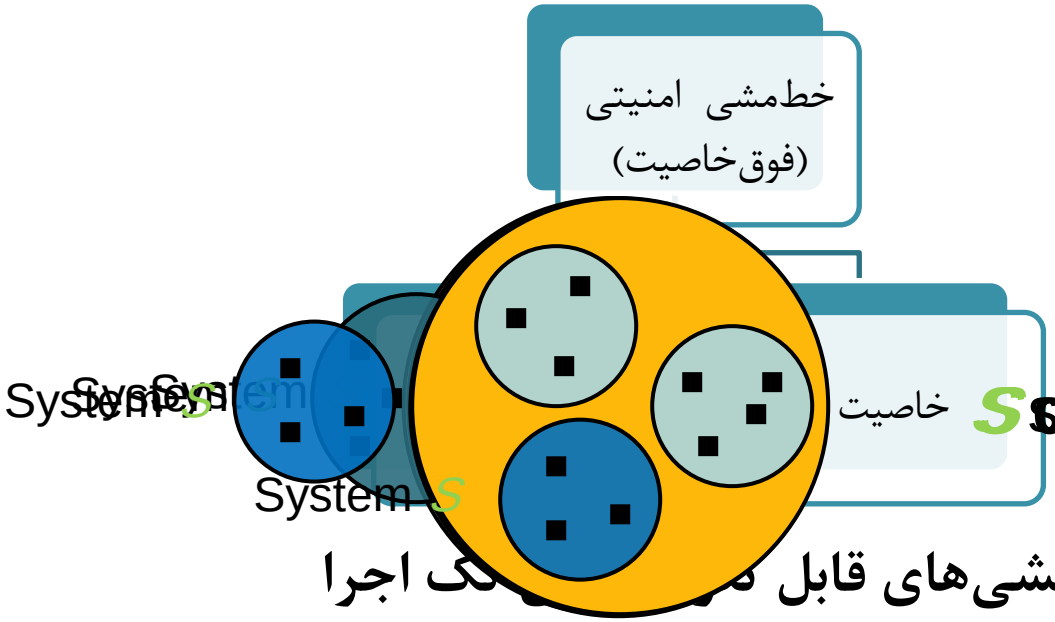
مقدمه



- خطمشی
  - محر
- تعریف
  - مقای
- خطمشی امنیتی عدم تداخل
  - بیان گزاره‌هایی روی اجراهای برنامه
  - انواع مختلف عدم تداخل

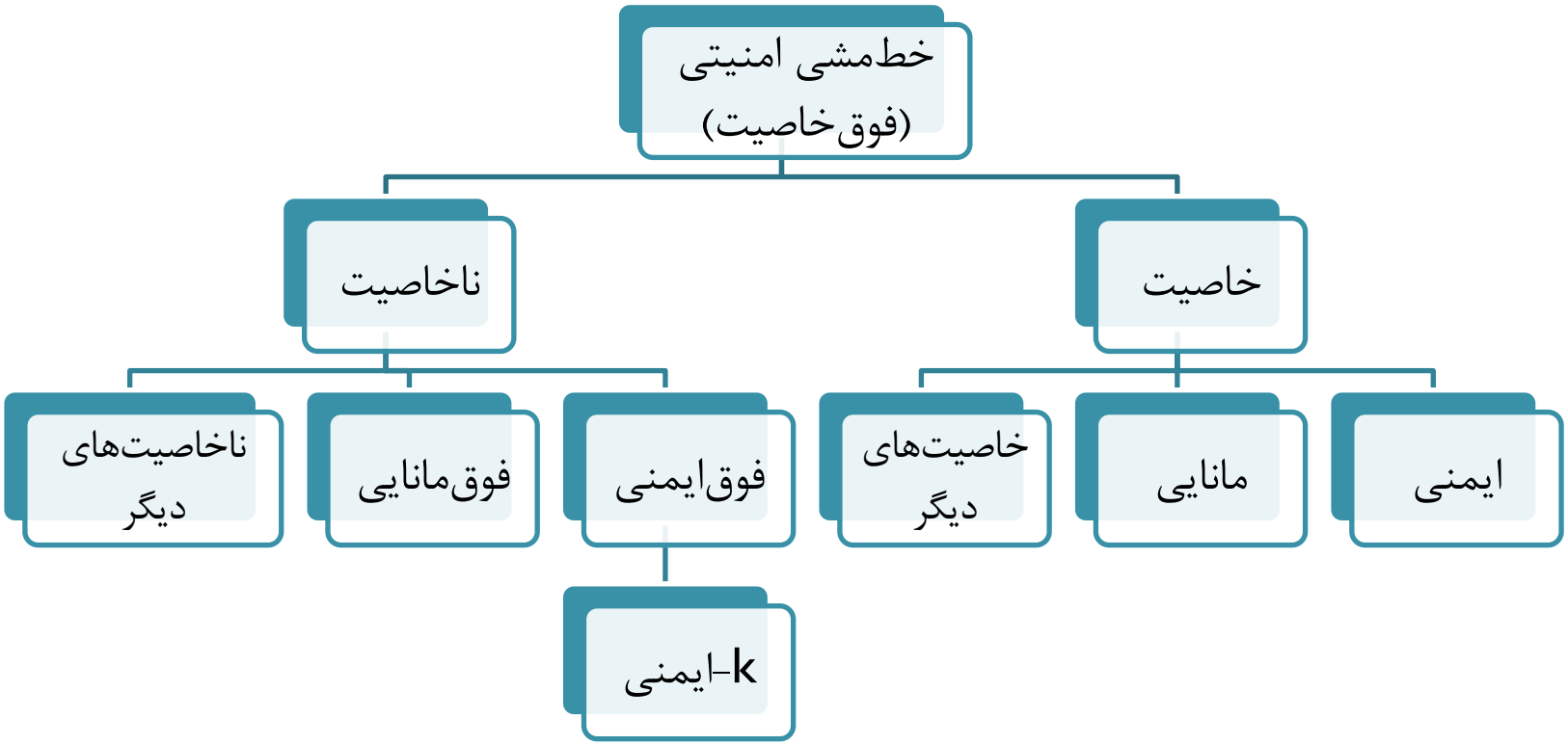
خط‌مشی امنیتی؛ خاصیت‌ها و فوق‌خاصیت‌ها

- تعریف صوری خط‌مشی امنیتی

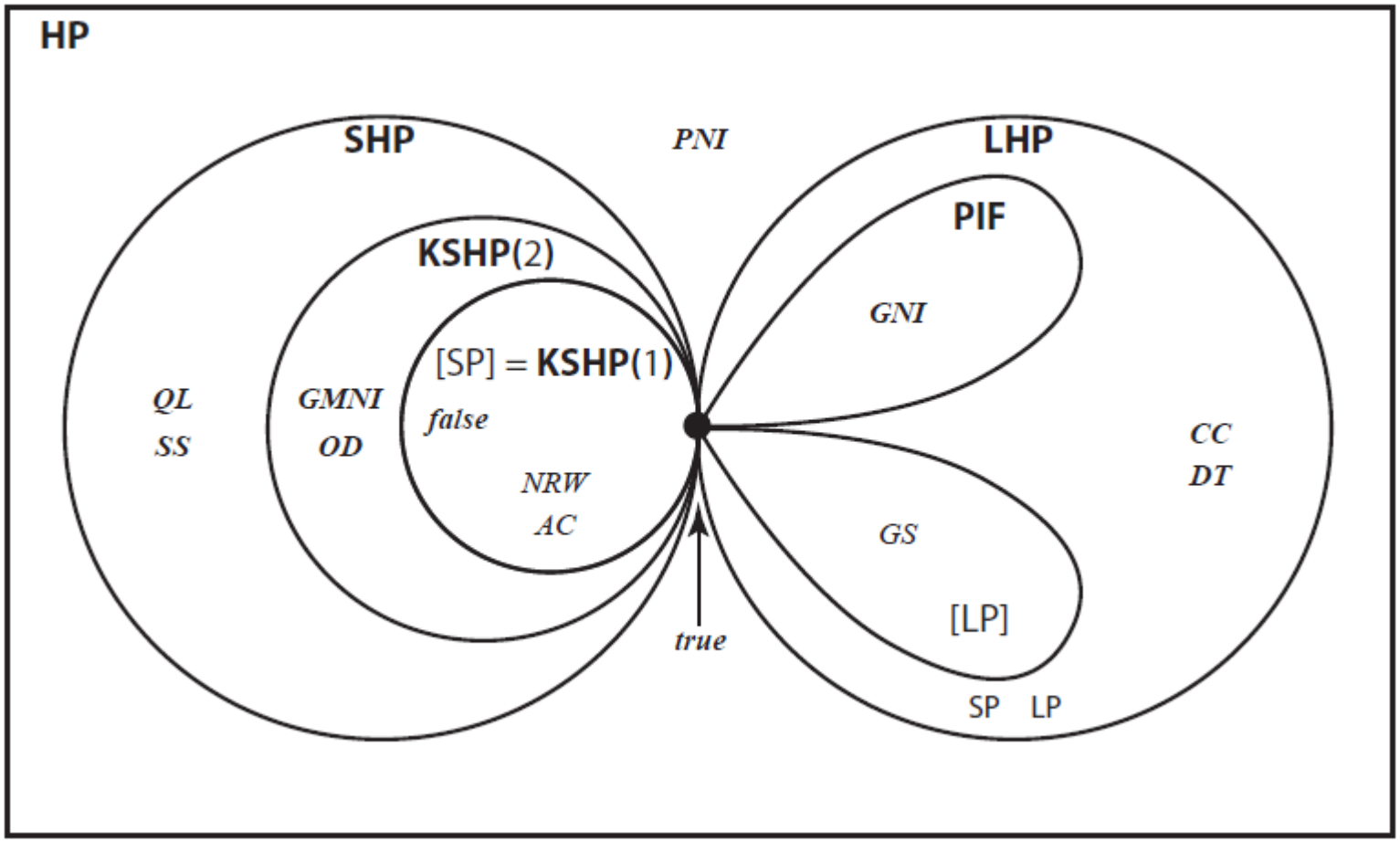


- خاصیت: خط‌مشی‌های قابل تعریف روی یک اجرا
  - مانند عدم خاتمه
- فوق‌خاصیت: خط‌مشی‌های قابل تعریف روی مجموعه‌ای از اجراها
  - ناخاصیت مانند جریان اطلاعات
    - = trace
    - = trace

خط‌مشی امنیتی؛ خاصیت‌ها و فوق‌خاصیت‌ها (ادامه)



خط‌مشی امنیتی؛ خاصیت‌ها و فوق‌خاصیت‌ها (ادامه)

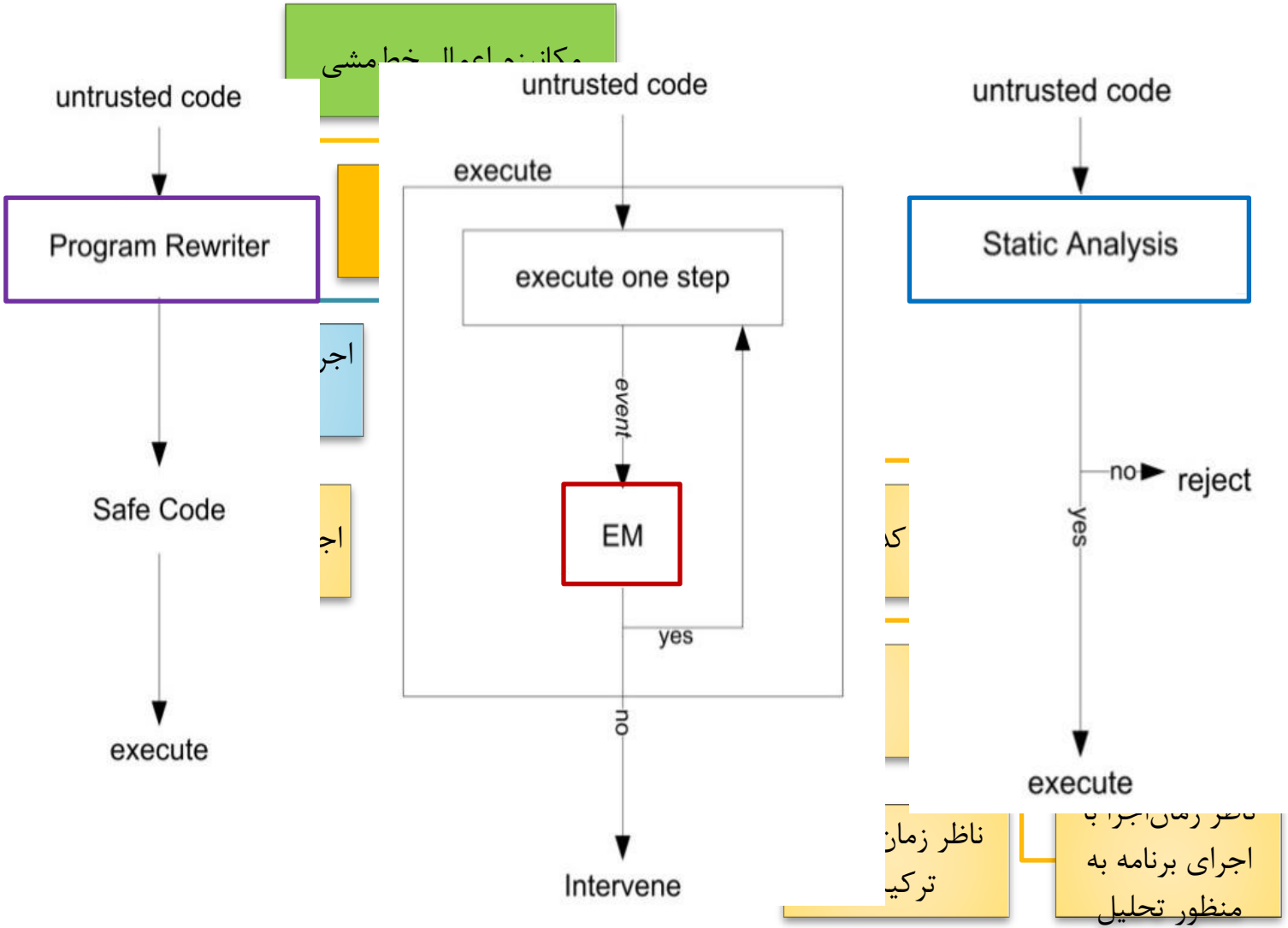


شکل ۱ – دسته‌بندی خط‌مشی‌های امنیتی [۴۷]

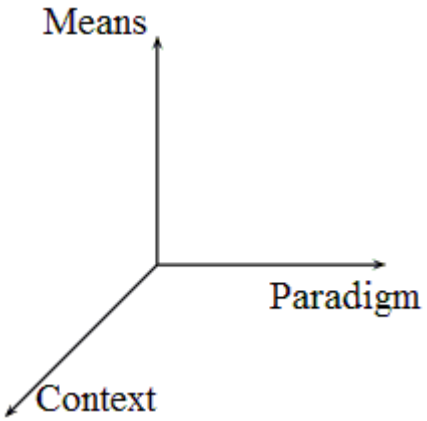
مکانیزم اعمال خط‌مشی امنیتی

- تعریف اعمال
  - وادارکردن مردم به اجرای قانون
  - اطمینان از برآورده‌شدن خط‌مشی در سامانه
- تعریف مکانیزم
  - روش، ابزار یا رویه‌ای برای اعمال خط‌مشی امنیتی
- معیارهای مقایسه
  - درستی
    - برنامه پس از اعمال توسط مکانیزم، واقعاً امن باشد.
  - شفافیت
    - برنامه امن توسط مکانیزم نیز امن شناخته شود.

دسته‌بندی مکانیزم‌های اعمال خط‌مشی



محورهای تأثیرگذار در توانایی ناظرها



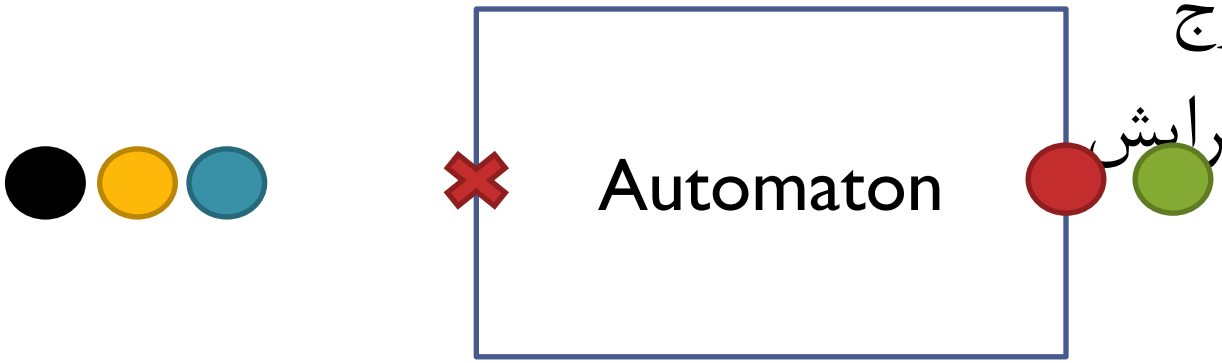
۱) وسیله در اختیار ناظر برای اعمال
- فقط قطع اجرای برنامه یا تغییر اجرا
  - بیان و مدل‌سازی به کمک خودکاره‌ها

۲) یکنواخت بودن یا نبودن زمینه
- دانش در خصوص رفتارهای ممکن برنامه
  - داشتن اطلاعات ایستا

۳) پارادایم اعمال
- قواعد کنترل‌کننده روش مجاز تبدیل اجراهای برنامه توسط ناظر
- میزان آزادی عمل ناظر در تبدیل اجرای داده‌شده

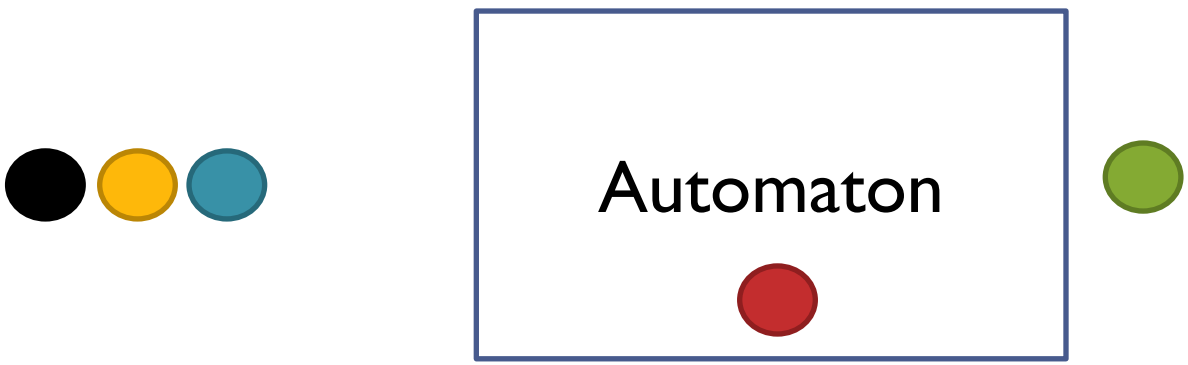
بیان و مدل‌سازی ناظرها به کمک خودکاره‌ها

- خودکاره قطع‌کننده
- خودکاره توقیف
- خودکاره درج
- خودکاره ویرایش



بیان و مدل‌سازی ناظرها به کمک خودکاره‌ها

- خودکاره توقیف



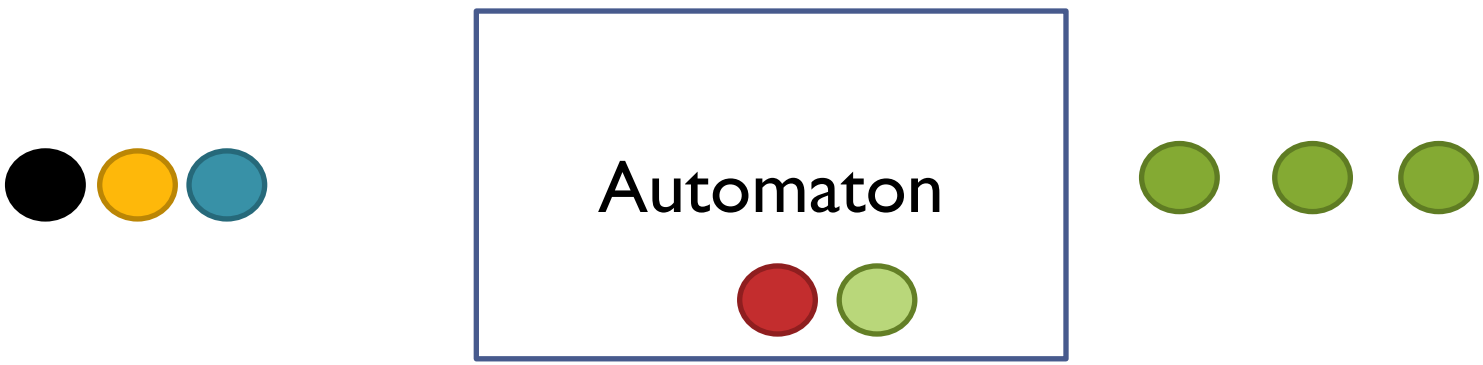
بیان و مدل‌سازی ناظرها به کمک خودکاره‌ها

- خودکاره درج

Automaton

بیان و مدل‌سازی ناظرها به کمک خودکاره‌ها

- خودکاره ویرایش
  - تلفیقی از خودکاره‌های توقیف و درج



پارادایم اعمال

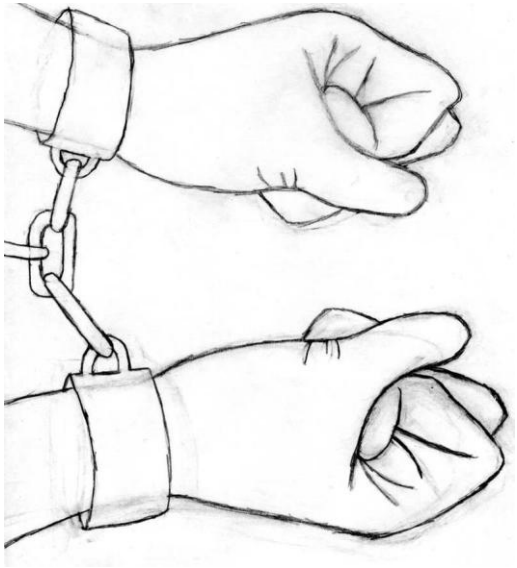
- اعمال دقیق
  - در هر گام از اجرا، کنش ورودی به عنوان بخشی از یک دنباله معتبر پذیرفته شود.
    - تساوی نحوی دنباله ورودی و خروجی، بلافاصله پس از مشاهده هر کنش!
- اعمال مؤثر
  - دنباله خروجی هم‌ارز دنباله ورودی باشد.
    - رابطه تساوی نحوی

مقایسه توانایی ناظرها در سه محور

- خودکاره قطع‌کننده + زمینه یکنواخت + پارادایم دقیق
  - دقیقاً مجموعه خاصیت‌های ایمنی
- خودکاره قطع‌کننده + زمینه غیریکنواخت + پارادایم دقیق
  - مجموعه خاصیت‌های ایمنی و بخشی از خاصیت‌های مانایی
- خودکاره قطع‌کننده + زمینه غیریکنواخت + پارادایم مؤثر
  - مجموعه‌ای بزرگ‌تر از مورد دوم

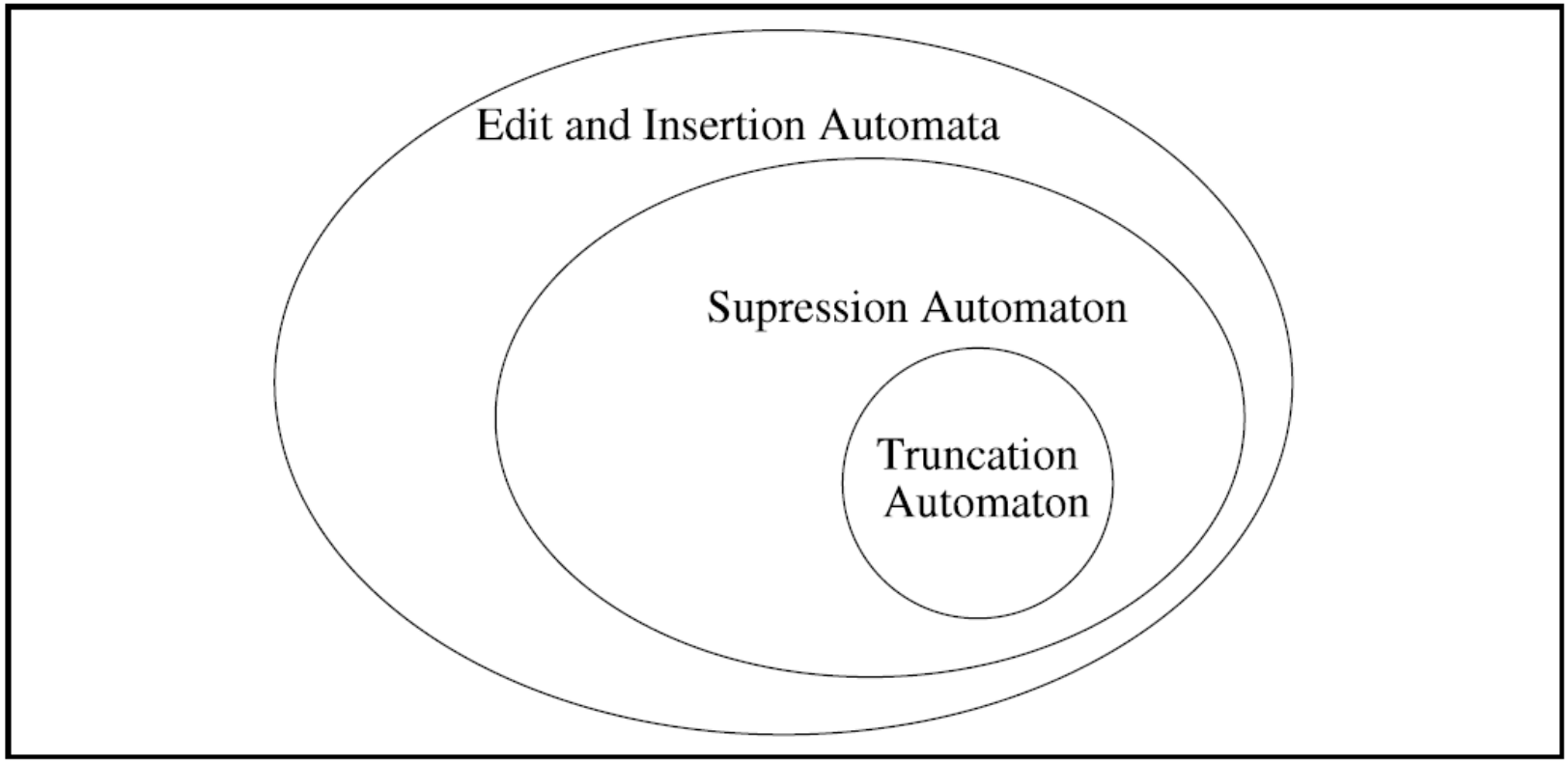
مقایسه توانایی ناظرها در سه محور (ادامه)

- زمینه یکنواخت + پارادایم دقیق
  - توانایی خودکاره قطع‌کننده با خودکاره‌های توقیف، درج و ویرایش برابر است = مجموعه خاصیت‌های ایمنی
    - طبق تعریف پارادایم دقیق!



مقایسه توانایی ناظرها در سه محور (ادامه)

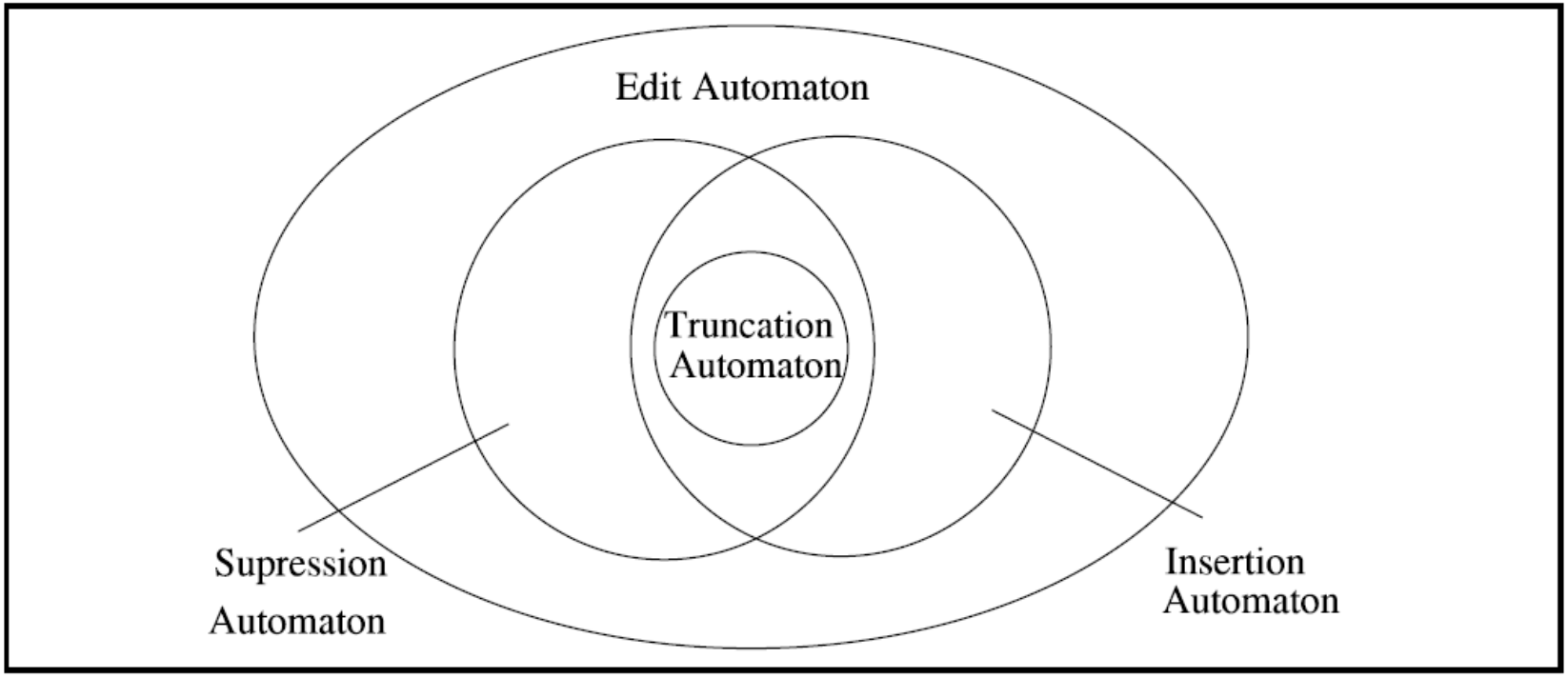
- زمینه غیریکنواخت + پارادایم دقیق



شکل ۲ – نمای گرافیکی قدرت خودکاره‌های مختلف در اعمال دقیق در محیط غیریکنواخت [۳۴]

مقایسه توانایی ناظرها در سه محور (ادامه)

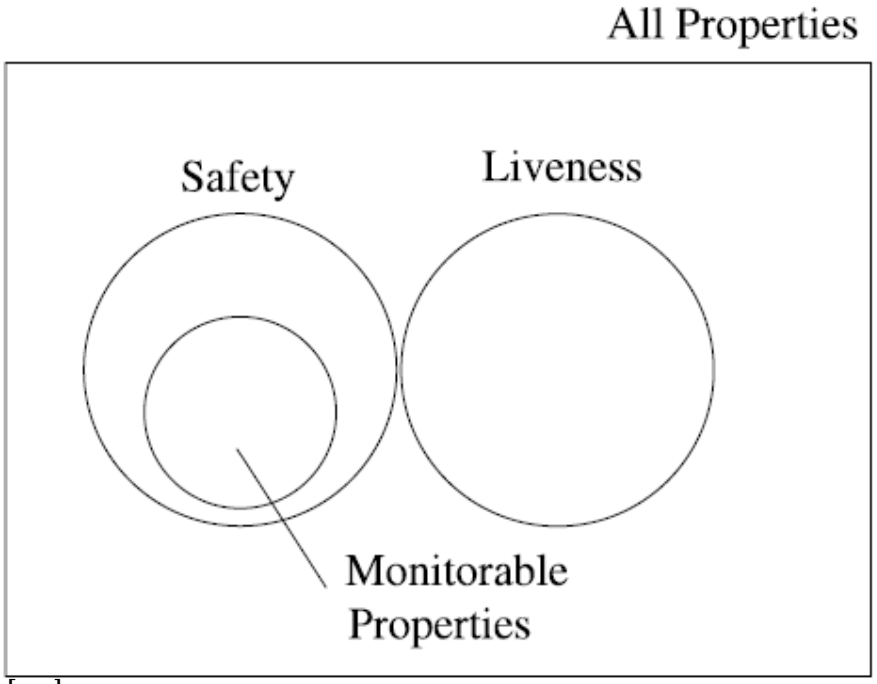
- زمینه یکنواخت + پارادایم مؤثر



شکل ۳ – نمای گرافیکی قدرت خودکاره‌های مختلف در اعمال مؤثر در محیط یکنواخت [۳۴]

تأثیر محدودیت‌های محاسباتی و حافظه‌ای در توانایی ناظرها

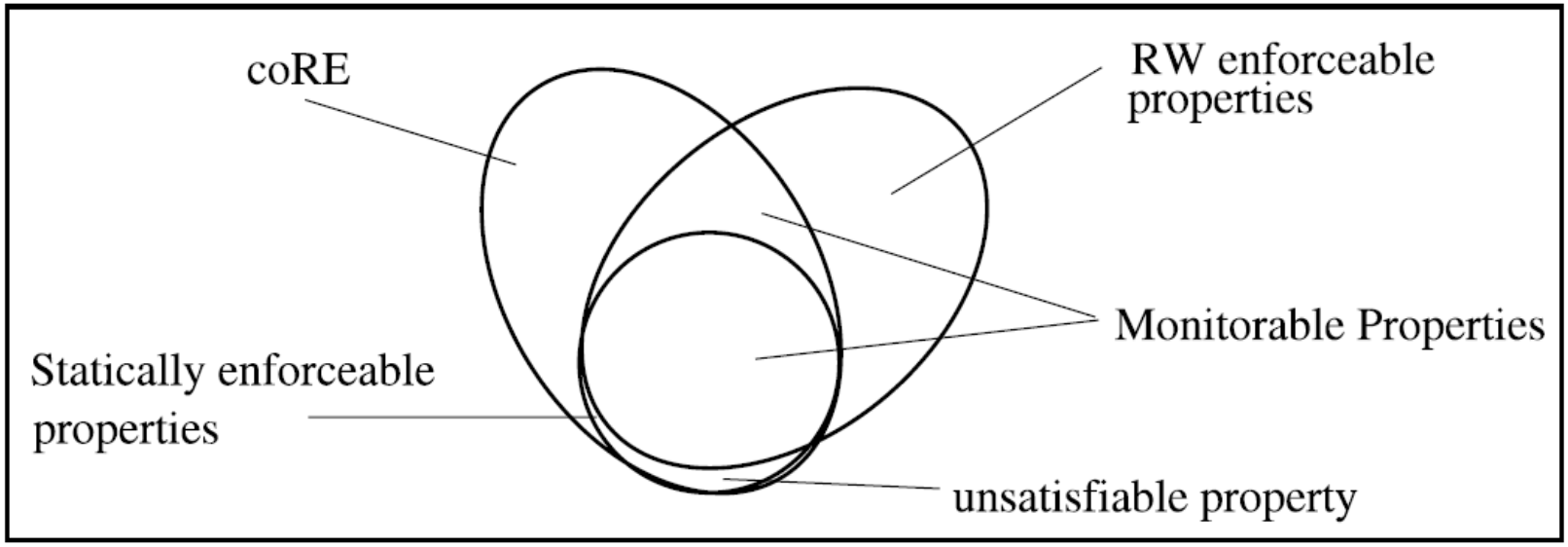
- محدودیت محاسباتی
  - فقط خاصیت‌های ایمنیِ مکمل شمارش‌پذیر بازگشتی توسط خودکاره امنیتی سنتی قابل اعمال است.



شکل ۴– خاصیت‌های قابل اعمال توسط خودکاره امنیتی سنتی [۳۸]

تأثیر محدودیت‌های محاسباتی و حافظه‌ای در توانایی ناظرها (ادامه)

- محدودیت محاسباتی



شکل ۵– خاصیت‌های قابل اعمال توسط مکانیزم‌های مختلف [۱۴]

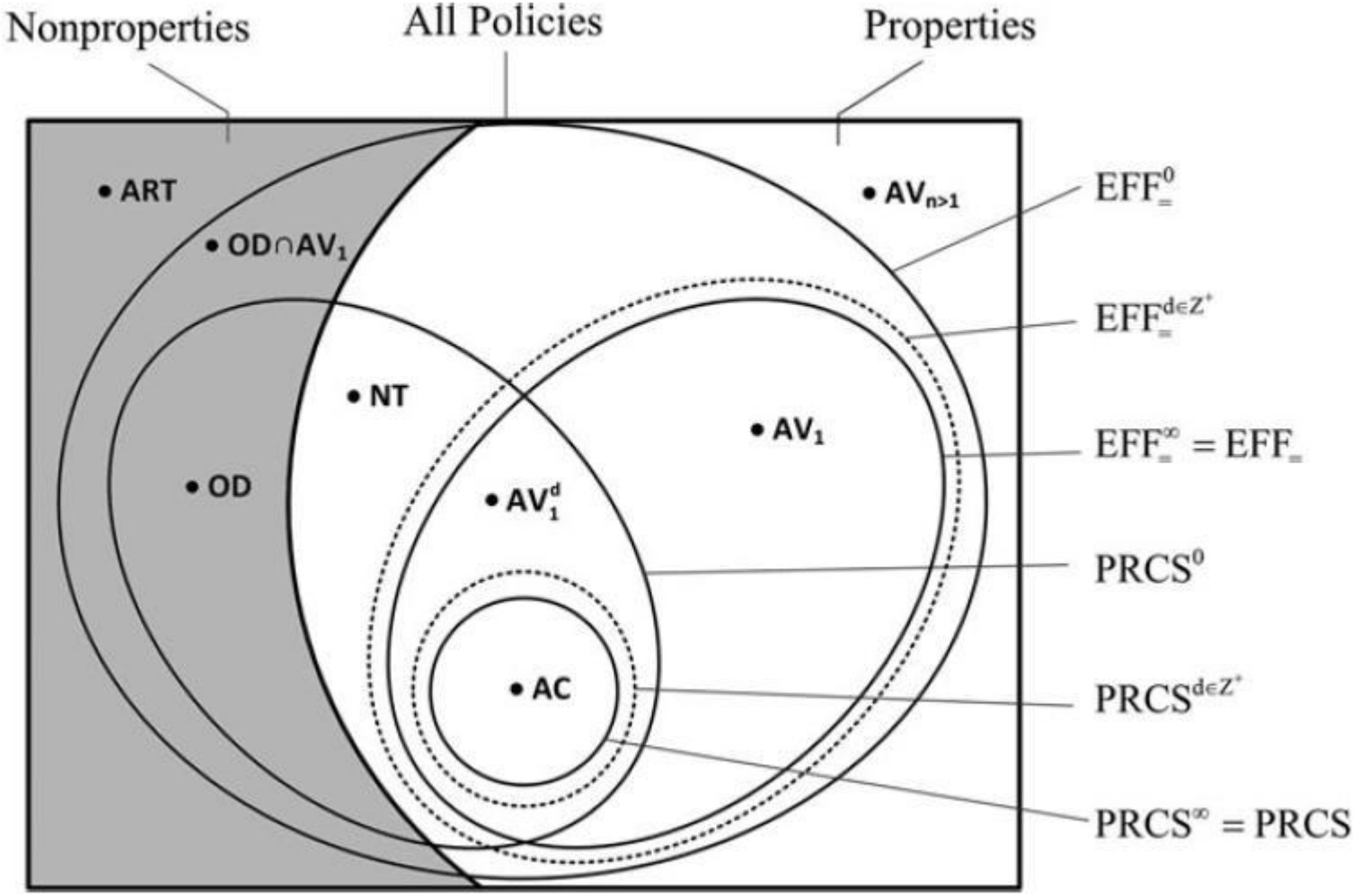
تأثیر محدودیت‌های محاسباتی و حافظه‌ای در توانایی ناظرها (ادامه)

- محدودیت حافظه‌ای
  - خودکاره تاریخچه کم‌عمق
    - زیرمجموعه محضی از مجموعه خاصیت‌های قابل‌اعمال توسط خودکاره امنیتی سنتی
  - خودکاره تاریخچه محدود
    - استفاده از حافظه‌ای محدود برای ذخیره‌سازی تاریخچه اجراهای برنامه
    - زیرمجموعه‌ای از خاصیت‌های ایمنی

پارادایم اعمال اصلاحی

- در پارادایم‌های قبلی، هیچ محدودیتی روی رفتار ناظر در برخورد با دنباله نامعتبر وجود نداشت.
  - تولید دنباله تهی!
- کمترین تغییرات ممکن نسبت به دنباله ورودی
  - هم‌ارزبودن معناشناختی دنباله خروجی به ازای همه دنباله‌های ورودی
    - هم سازگار با خط‌مشی و هم سازگار با رفتار اولیه برنامه
- حالت خاص: اعمال مؤثر تحت رابطه تساوی
- استفاده از تعابیر دیگر برای مفهوم پارادایم اعمال

تأثیر در اختیار داشتن اطلاعات ایستا در توانایی ناظرها



شکل ۶– ارتباط بین خط‌مشی‌های قابل اعمال در پارادایم‌های اعمال مختلف [۴۶]

مقایسه تکنیک‌های اعمال زمان‌اجرا

- مداخله فراخوانی سامانه
- مفسر ایمن
- ایزوله‌کردن خطای نرم‌افزار
- ناظرهای مرجع درخط
- ترجمه پویای نرم‌افزار
- ردیاب‌های جریان داده

مقایسه تکنیک‌های اعمال زمان‌اجرا (ادامه)

- مداخله فراخوانی سامانه
  - مداخله‌گری به ازای هر دستور یا فراخوانی سامانه
  - بررسی وقوع فراخوانی‌های خاصی توسط ناظر
    - مسدودکردن یا تغییردادن

Criterion	System call implementations
Abstraction level	Operating system
Guarantees	mediation (full if in kernel), tamper-proof
Trusted components	OS, system call wrappers, libraries
Policy class	access control
Policy language	very low level, not user-friendly
Overheads	very low (ms), depends on no.system calls

مقایسه تکنیک‌های اعمال زمان‌اجرا (ادامه)

- مفسر ایمن
  - وجود یک لایه مجازی میانجی بین برنامه در حال اجرا و پردازنده
  - استفاده از تفسیر کد

Criterion	Safe interpreter implementations
Abstraction level	Application level
Guarantees	tamper-proof, but not non-bypassability
Trusted components	browser, helper modules,interpreter
Policy class	access control
Policy language	scripting languages, or customized
Overheads	from 1 to 25% (peaks at 200%)

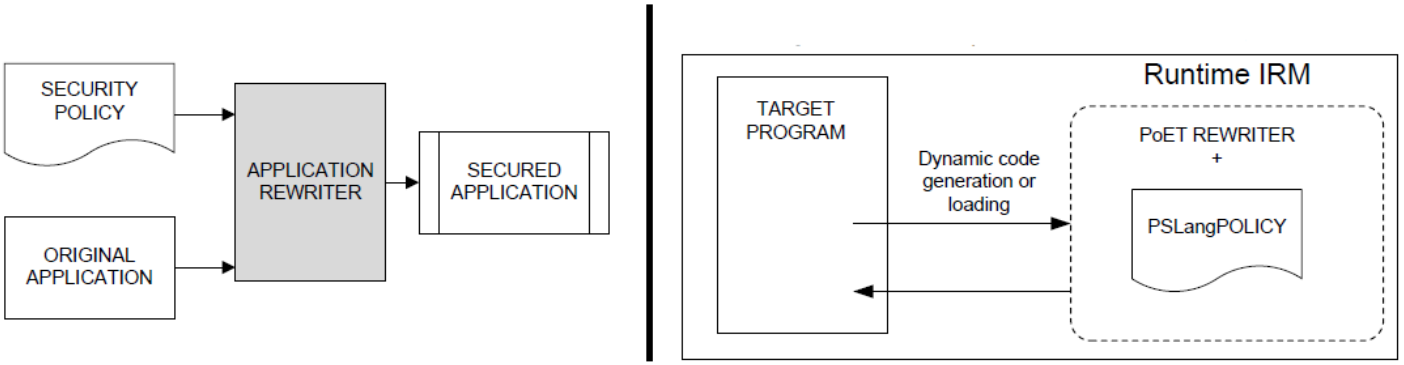
مقایسه تکنیک‌های اعمال زمان‌اجرا (ادامه)

- ایزوله‌کردن خطای نرم‌افزار
  - تغییر آدرس‌های حافظه در کد شی یا اسمبلر
    - جلوگیری از خواندن، نوشتن و پرش به آدرس‌های خارج از ناحیه تعریف‌شده توسط خط‌مشی
  - در زمان بارگذاری، و نه زمان اجرا

Criteria	SFI implementations
Abstraction	OS and platform
Guarantees	nonbypassability, tamproofness
Trust components	OS, rewriter, compiler
Policy class	access control
Policy language	high-level (Naccio) otherwise very low level
Overheads	low to medium, e.g., 9-45%

مقایسه تکنیک‌های اعمال زمان‌اجرا (ادامه)

- ناظر مرجع درخط
  - استفاده از یک ابزار بازنویس برای درج کدهای امنیتی در برنامه



Criteria	IRM implementations
Abstraction	application and platform
Guarantees	mediation, integrity, tamproofness
Trust components	rewriter, policy compiler
Policy class	access control
Policy language	security automata or Java-like
Overheads	low to medium, e.g., 0.1-30%

مقایسه تکنیک‌های اعمال زمان‌اجرا (ادامه)

- ترجمه پویای نرم‌افزار
  - تبدیل خط‌به‌خط کد کامپایل‌شده به کد دودویی امن و بهینه
    - مفسر ایمن اجرای یک دستور را تبدیل می‌کرد.
    - مثال: ماشین‌های مجازی

Criteria	SDT implementations
Abstraction level	runtime and application logic level
Guarantees	nonbypassable, tamperproof
Trusted comp.	OS,interpreter,compiler
Policy class	access control
Policy language	low-level, or custom
Overheads	low to medium e.g., 2-30%

مقایسه تکنیک‌های اعمال زمان‌اجرا (ادامه)

- ردیاب‌های جریان داده
  - برای اعمال خط‌مشی‌های کنترل جریان اطلاعات
  - استفاده از تحلیل‌های ایستا و پویا
    - جلوگیری از حملات تزریق
    - ابزارهایی مشابه TaintDroid و RIFLE

Criterion	Data Flow Trackers
Abstraction level	OS, runtime, platform, application
Guarantees	full mediation but not always proven
Policy class	both access and usage control
Trust components	the OS, runtime or platform libraries
Policy language	complex, Java-like syntax in best case
Overheads	very high with minimum overheads around 30%

مقایسه تکنیک‌های اعمال زمان‌اجرا (ادامه)

- مقایسه از نظر میزان سربار
  - هرچه از سطح ماشین و سیستم‌عامل دورتر، سربار بیشتر
    - مداخله‌گری فراخوانی‌های سامانه؛ کم‌ترین سربار
    - مفسرهای ایمن بین ۱ تا ۲۵ درصد
    - ایزوله‌کردن خطای نرم‌افزار بین ۹ تا ۴۵ درصد
    - ناظرهای مرجع درخط بین ۰.۱ تا ۳۰ درصد
    - مترجم‌های پویای نرم‌افزار بین ۲ تا ۳۰ درصد
    - ردیاب‌های جریان داده بیش از ۳۰ درصد

مقایسه مکانیزم‌های پویای اعمال جریان اطلاعات

- عدم تداخل
  - غیرحساس به خاتمه (TINI)
    - اجازه وقوع کانال‌های خاتمه
  - آگاه به خاتمه (TANI)
    - ناظر کانال خاتمه جدیدی اضافه نکند.
  - حساس به خاتمه (TSNI)
    - جلوگیری از وقوع هرگونه کانال خاتمه

مقایسه مکانیزم‌های پویای اعمال جریان اطلاعات (ادامه)

- ارتقای بدون حساسیت
  - اجازه هیچ به‌روزرسانی‌ای به متغیرهای سطح پایین در زمینه امنیتی سطح بالا داده نمی‌شود.
    - نمونه‌ای زمینه امنیتی سطح بالا، شاخه‌های شرط با عبارت شرطی وابسته به متغیر سطح بالا
  - اعمال فقط عدم تداخل غیرحساس به خاتمه

مقایسه مکانیزم‌های پویای اعمال جریان اطلاعات (ادامه)

- ارتقای آسان‌گیر
  - برچسب‌گذاری متغیر سطح پایین به‌روزشده در زمینه سطح بالا
    - توقف اجرا در صورتی‌که در ادامه اجرا، انشعابی باشد که به متغیر برچسب‌دار وابسته است.
  - اعمال فقط عدم تداخل غیرحساس به خاتمه

مقایسه مکانیزم‌های پویای اعمال جریان اطلاعات (ادامه)

- ناظر ترکیبی
  - ترکیب تحلیل ایستا و پویا
    - تحلیل ایستای شاخه‌هایی از شرط که اجرا نمی‌شوند.
    - ارتقای سطح امنیتی متغیرها به سطح عبارت شرطی
  - اعمال عدم تداخل آگاه به خاتمه

مقایسه مکانیزم‌های پویای اعمال جریان اطلاعات (ادامه)

- چنداجرایی امن
  - اجرای هم‌زمان برنامه به ازای هر سطح امنیتی
    - در هر سطح فقط ورودی‌های قابل مشاهده آن سطح داده می‌شود و برای متغیرهای ورودی سطح بالاتر، مقداری ثابت در نظر گرفته می‌شود.
    - حذف خروجی‌های سطح پایین، برای جلوگیری از تکرار
  - استفاده از استراتژی زمان‌بندی
    - نیازمند تغییر محیط اجرا
  - اعمال عدم تداخل حساس به خاتمه

مقایسه مکانیزم‌های پویای اعمال جریان اطلاعات

- ارتقای بدون حساسیت
- ارتقای آسان‌گیر
- ناظر ترکیبی
- اجرای چندباره امن

	NSU	PU	HM	SME	MF
NSU		$\not\supseteq_{\mathcal{P}} \not\supseteq_{\mathcal{F}}$	$\not\supseteq_{\mathcal{P}} \supseteq_{\mathcal{F}}$	$\not\supseteq_{\mathcal{P}} \not\supseteq_{\mathcal{F}}$	$\not\supseteq_{\mathcal{P}} \not\supseteq_{\mathcal{F}}$
PU	$\supseteq_{\mathcal{T}}$ $\supseteq_{\mathcal{F}}$		$\not\supseteq_{\mathcal{P}} \supseteq_{\mathcal{F}}$	$\not\supseteq_{\mathcal{P}} \not\supseteq_{\mathcal{F}}$	$\not\supseteq_{\mathcal{P}} \not\supseteq_{\mathcal{F}}$
HM	$\supseteq^*_{\mathcal{P}} \not\supseteq_{\mathcal{F}}$	$\supseteq^*_{\mathcal{P}} \not\supseteq_{\mathcal{F}}$		$\not\supseteq_{\mathcal{P}} \not\supseteq_{\mathcal{F}}$	$\not\supseteq_{\mathcal{P}} \not\supseteq_{\mathcal{F}}$
SME	$\supseteq^*_{\mathcal{P}} \not\supseteq_{\mathcal{F}}$	$\supseteq^*_{\mathcal{P}} \not\supseteq_{\mathcal{F}}$	$\supseteq^*_{\mathcal{P}} \supseteq_{\mathcal{F}}$		$\supseteq^*_{\mathcal{P}} \not\supseteq_{\mathcal{F}}$
MF	$\supseteq_{\mathcal{T}} \supseteq_{\mathcal{F}}$	$\supseteq_{\mathcal{T}} \supseteq_{\mathcal{F}}$	$\not\supseteq_{\mathcal{P}} \supseteq_{\mathcal{F}}$	$\not\supseteq_{\mathcal{P}} \not\supseteq_{\mathcal{F}}$	

$\supseteq_{\mathcal{T}}$ more true TINI transparent than

$\supseteq_{\mathcal{P}}$ more TINI precise than ($\not\supseteq_{\mathcal{P}} \implies \not\supseteq_{\mathcal{T}}$)

$\supseteq^*_{\mathcal{P}}$ more TSNI precise than

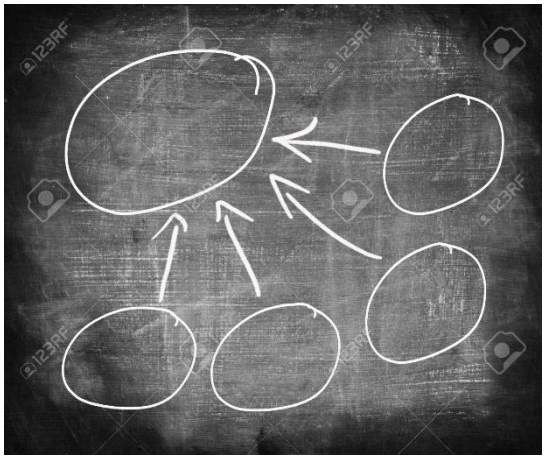
$\supseteq_{\mathcal{F}}$ more false TINI transparent than

Monitor is TANI

Monitor is TSNI, hence TANI

جمع‌بندی

- تعریف ناخاصیت جریان اطلاعات
- بررسی انواع مکانیزم‌های اعمال خط‌مشی
- بررسی مکانیزم‌های نظارت زمان‌اجرا و توانایی آن‌ها در شرایط و پارادایم‌های مختلف
- بررسی تکنیک‌ها و مکانیزم‌های پویای دیگر



مسائل باز

- سرشت‌نمایی خط‌مشی‌های قابل اعمال توسط مکانیزم‌های مختلف
  - میزان اطلاعات لازم برای اعمال یک خط‌مشی
- تغییر خط‌مشی در طول اجرای برنامه
- ارائه معیار مقایسه بین مکانیزم‌های مختلف
- ارائه و بهبود مکانیزم‌های موجود
  - افزایش شفافیت
  - کاهش سربار

پروژه کارشناسی ارشد

- بهبود مکانیزم‌های مبتنی بر اجرای چندباره برای اعمال خط‌مشی‌های جریان اطلاعات
  - مطالعه فوق‌خاصیت‌های جریان اطلاعات
  - مطالعه انواع مکانیزم‌های پویا، مانند نظارت زمان‌اجرا، تحلیل ترکیبی و اجرای چندباره امن
  - ارائه و بیان صوری خط‌مشی جریان اطلاعات و زبان برنامه‌نویسی موردنظر
    - سطح انتزاع و موازنه کاربردی‌بودن-مدل‌بودن
  - ارائه مکانیزم مبتنی بر اجرای چندباره (با استفاده از ایده خودترکیبی)
  - اثبات درستی و شفافیت مکانیزم ارائه‌شده و مقایسه با مکانیزم‌های موجود
  - نگارش پایان‌نامه

منابع و مراجع

[1] J. McLean, "A general theory of composition for trace sets closed under selective interleaving functions," *Res. Secur. Privacy, 1994. Proceedings., 1994 IEEE Comput. Soc. Symp.*, pp. 79–93, 1994.

[2] F. B. Schneider, "Enforceable security policies," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 1, pp. 30–50, 2000.

[3] M. Bishop, *Computer Security: Art and Science*, 2nd ed. Addison-Wesley, 2003.

[4] G. Barthe, J. M. Crespo, D. Devriese, F. Piessens, and E. Rivas, "Secure multi-execution through static program transformation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, vol. 7273 LNCS, pp. 186–202.

[5] E. Cohen, "Information Transmission in Computational Systems," *Proc. Sixth ACM Symp. Oper. Syst. Princ.*, no. November, pp. 133–139, 1977.

[6] G. Le Guernic, "Confidentiality enforcement using dynamic information flow analyses," PhD Thesis, Kansas State University, 2007.

[7] J. A. Goguen and J. Meseguer, "Security Policies and Security Models," *Secur. Privacy, IEEE Symp.*, vol. 0, p. 11+, 1982.

[8] J. McLean, "Security models and information flow," *Res. Secur. Privacy, 1990. Proceedings., 1990 IEEE Comput. Soc. Symp.*, pp. 180–187, 1990.

[9] D. Sutherland, "A model of information," in *Proc. 9th National Computer Security Conference*, 1986, pp. 175–183.

[10] A. Sabelfeld and A. C. Myers, "Language-based information-flow security," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 1, pp. 5–19, 2003.

منابع و مراجع

[11] D. E. Denning, "A lattice model of secure information flow," *Commun. ACM*, vol. 19, no. 5, pp. 236–243, 1976.

[12] D. Hedin and A. Sabelfeld, "A perspective on information-flow control," in *NATO Science for Peace and Security Series - D: Information and Communication Security*, vol. 33: Softwa, no. 10.3233/978-1-61499-028-4-319, 2012, pp. 319–347.

[13] S. Hunt and D. Sands, "On flow-sensitive security types," *ACM SIGPLAN Not.*, vol. 41, no. 1, pp. 79–90, 2006.

[14] K.W. Hamlen, F. B. Schneider, K.W. Hamlen, F. B. Schneider, and G. Morrisett, "Computability Classes for Enforcement Mechanisms," *ACM Trans. Program. Lang. Syst.*, vol. 28, no. 1, pp. 175–205, 2006.

[15] D. Volpano, C. Irvine, and G. Smith, "A sound type system for secure flow analysis," *J. Comput. Secur.*, vol. 4, no. 2/3, p. 167, 1996.

[16] G. Barthe, P. R. D'Argenio, and T. Rezk, "Secure information flow by self-composition," *Proceedings. 17th IEEE Comput. Secur. Found. Work. 2004.*, pp. 1–52, 2004.

[17] J. Ligatti and S. Reddy, "A theory of runtime enforcement, with results," in *European Symposium on Research in Computer Security*, 2010, pp. 87–100.

[18] D. Devriese and F. Piessens, "Noninterference through secure multi-execution," in *Proceedings - IEEE Symposium on Security and Privacy*, 2010, pp. 109–124.

[19] J. Ligatti, L. Bauer, and D. Walker, "Enforcing non-safety security policies with program monitors," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2005, vol. 3679 LNCS, pp. 355–373.

[20] C. E. Irvine, "The reference monitor concept as a unifying principle in computer security education," Technical Report, NAVAL Postgraduate Sschool Monterey CA Dept of Computer Science, 1999.

منابع و مراجع

[21] Ú. Erlingsson, "The inlined reference monitor approach to security policy enforcement," phdthesis, Cornell University, 2004.

[22] M. Viswanathan, "Foundations for the Run-time Analysis of Software systems," PhD Thesis, University of Pennsylvania, 2000.

[23] G. Gheorghe and B. Crispo, "A survey of runtime policy enforcement techniques and implementations," University of Trento, Technical Report # DISI-11-477, 2011.

[24] A. Sabelfeld and A. Russo, "From dynamic to static and back: riding the roller coaster of Information-flow control research," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, vol. 5947 LNCS, pp. 352–365.

[25] L. Zheng and A. C. Myers, "Dynamic security labels and static information flow control," in *International Journal of Information Security*, 2007, vol. 6, no. 2–3, pp. 67–84.

[26] N. Broberg and D. Sands, "Flow-sensitive semantics for dynamic information flow policies," *Proc. ACM SIGPLAN Fourth Work. Program. Lang. Anal. Secur. - PLAS '09*, p. 101, 2009.

[27] P. Shroff, S. F. Smith, and M. Thober, "Dynamic dependency monitoring to secure information flow," in *Proceedings - IEEE Computer Security Foundations Symposium*, 2007, pp. 203–217.

[28] D. E. Denning and P. J. Denning, "Certification of programs for secure information flow," *Commun. ACM*, vol. 20, pp. 504–513, 1977.

[29] G. Smith and D. Volpano, "Secure information flow in a multi-threaded imperative language," in *Proceedings of the 25th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, 1998, pp. 355–364.

[30] D. Volpano and G. Smith, "Probabilistic noninterference in a concurrent language," in *Proceedings of the Computer Security Foundations Workshop*, 1998, pp. 34–43.

منابع و مراجع

[31] A. Sabelfeld and D. Sands, "Probabilistic noninterference for multi-threaded programs," *Proc. 13th IEEE Comput. Secur. Found. Work. CSFW-13*, pp. 200–214, 2000.

[32] S. A. Zdancewic, "Programming languages for information security," phdthesis, Cornell University, 2002.

[33] J. Ligatti, L. Bauer, and D. Walker, "Run-Time Enforcement of Nonsafety Policies," *ACM Trans. Inf. Syst. Secur.*, vol. 12, no. 3, pp. 1–41, 2009.

[34] L. Bauer, J. Ligatti, and D. Walker, "More enforceable security policies," in *Proceedings of the Workshop on Foundations of Computer Security (FCS02), Copenhagen, Denmark*, 2002.

[35] J. Ligatti, L. Bauer, and D. Walker, "Edit automata: Enforcement mechanisms for run-time security policies," *Int. J. Inf. Secur.*, vol. 4, no. 1–2, pp. 2–16, 2005.

[36] C. Talhi, N. Tawbi, and M. Debbabi, "Execution monitoring enforcement under memory-limitation constraints," *Inf. Comput.*, vol. 206, no. 2–4, pp. 158–184, 2008.

[37] R. Khoury and N. Tawbi, "Which security policies are enforceable by runtime monitors? A survey," *Computer Science Review*, vol. 6, no. 1. pp. 27–45, 2012.

[38] M. Kim, S. Kannan, I. Lee, O. Sokolsky, and M. Viswanathan, "Computational analysis of run-time monitoring: Fundamentals of java-MaC," in *Electronic Notes in Theoretical Computer Science*, 2002, vol. 70, no. 4, pp. 85–99.

[39] A. Lamei, "Formal Characterization of Security Policy Enforcement through Program Rewriting," PhD Thesis, Amirkabir University of Technology, 2016.

[40] P.W. L. Fong, "Access control by tracking shallow execution history," in *Proceedings - IEEE Symposium on Security and Privacy*, 2004, vol. 2004, pp. 43–55.

منابع و مراجع

[41] D. Beauquier, J. Cohen, and R. Lanotte, "Security policies enforcement using finite and pushdown edit automata," *Int. J. Inf. Secur.*, vol. 12, no. 4, pp. 319–336, 2013.

[42] N. Bielova and F. Massacci, "Do you really mean what you actually enforced?," *Int. J. Inf. Secur.*, vol. 10, no. 4, pp. 239–254, 2011.

[43] R. Khoury and N. Tawbi, "Corrective enforcement of security policies," in *International Workshop on Formal Aspects in Security and Trust*, 2010, pp. 176–190.

[44] R. Khoury and N. Tawbi, "Using equivalence relations for corrective enforcement of security policies," in *International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security*, 2010, pp. 139–154.

[45] H. Chabot, R. Khoury, and N. Tawbi, "Extending the enforcement power of truncation monitors using static analysis," *Comput. Secur.*, vol. 30, no. 4, pp. 194–207, 2011.

[46] F. Imanimehr and M. S. Fallah, "How Powerful Are Run-Time Monitors with Static Information?," *The Computer Journal*, 2016.

[47] M. R. Clarkson and F. B. Schneider, "Hyperproperties," *J. Comput. Secur.*, vol. 18, no. 6, pp. 1157–1210, 2010.

[48] R. Khoury and N. Tawbi, "Corrective enforcement: a new paradigm of security policy enforcement by monitors," *ACM Trans. Inf. Syst. Secur.*, vol. 15, no. 2, p. 10, 2012.

[49] F. B. Schneider, M. Ngo, F. Massacci, D. Milushev, and F. Piessens, "Runtime Enforcement of Security Policies on Black Box Reactive Programs," *ACM SIGPLAN-SIGACT Symp. Princ. Program. Lang.*, vol. 3, no. 1, pp. 43–54, 2015.

[50] S. Zdancewic and A. C. Myers, "Observational determinism for concurrent program security," in *Computer Security Foundations Workshop, 2003. Proceedings. 16th IEEE*, 2003, pp. 29–43.

منابع و مراجع

[51] F. B. Schneider, G. Morrisett, and R. Harper, "A language-based approach to security," in *Informatics*, 2001, pp. 86–101.

[52] G. Le Guernic, A. Banerjee, T. Jensen, and D. A. Schmidt, "Automata-based confidentiality monitoring," in *Annual Asian Computing Science Conference*, 2006, pp. 75–89.

[53] A. Askarov, S. Hunt, A. Sabelfeld, and D. Sands, "Termination-insensitive noninterference leaks more than just a bit," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2008, vol. 5283 LNCS, pp. 333–348.

[54] N. Bielova and T. Rezk, "A taxonomy of information flow monitors," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9635, pp. 46–67.

[55] A. Russo and A. Sabelfeld, "Dynamic vs. static flow-sensitive security analysis," in *Proceedings - IEEE Computer Security Foundations Symposium*, 2010, pp. 186–199.

[56] T. H. Austin and C. Flanagan, "Multiple facets for dynamic information flow," *Proc. 39th Annu. ACM SIGPLAN-SIGACT Symp. Princ. Program. Lang. - POPL '12*, vol. 47, no. 1, p. 165, 2012.

[57] T. H. Austin and C. Flanagan, "Permissive dynamic information flow analysis," in *Proceedings of the 5th ACM SIGPLAN Workshop on Programming Languages and Analysis for Security - PLAS '10*, 2010, p. 3.

[58] D. Zanarini, M. Jaskelioff, and A. Russo, "Precise enforcement of confidentiality for reactive systems," in *Proceedings of the Computer Security Foundations Workshop*, 2013, pp. 18–32.

با سپاس از توجه شما! ☺