System and networking for portable objects proved to be safe

Isabelle Simplot-Ryl

DGA–INRIA seminary

June 2008

Outline

1. **Presentation of POPS project-team**
2. Communications in sensor networks
   - Overview of POPS activity in sensor networks
   - Example of key predistribution
3. Safety and security verifications in autonomous systems
   - General scheme for embedded verification
   - Applications/Experimental Results
4. Conclusion

POPS members

Head

- David Simplot-Ryl (Professor, Univ. Lille 1)

Members

- Jean Carle (Associate Professor, IUT-A, Univ. Lille 1)
- Gilles Grimaud (Associate Professor, Univ. Lille 1)
- Michaël Hauspie (Associate Professor, IUT-A, Univ. Lille 1)
- Samuel Hym (Associate Professor, Univ. Lille 1)
- Nathalie Mitton (Research officer, INRIA)
- Isabelle Simplot-Ryl (Professor, IUT-A, Univ. Lille 1)

In september ?

- Tahiry Razafindralambo (Research officer, INRIA)
- Marie-Emilie Voge (Associate Professor, Univ. Lille 1)

POPS members

Administrative staff

- Anne Rejl (Project assistant, INRIA)

External collaborators

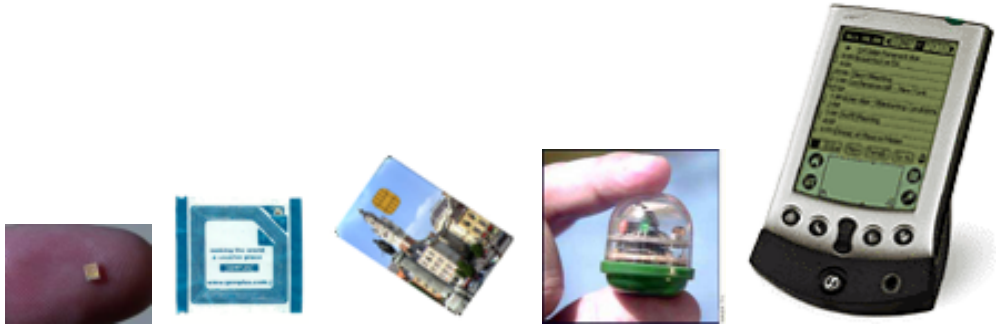
- Ivan Stojmenovic (Professor, Univ. Ottawa, Canada)
- Issa Traoré (Associate Professor, Univ. Victoria, Canada)
- Jean-Jacques Vandewalle (Researcher, Gemplus)

Temporary staff

- 6 PhD students, 1 engineer, masters, interships, ...

Technological context

POPS = Tiny targets = Constrained Hardware



POPS=Portable Objects Proved to be Safe

- Technical characteristics:
  - From 8 to 32 bits Processors
  - 1Kb of RAM, 64 Kb of E^2PROM, 128 Kb ROM
  - Limited electrical resources
  - Unsafe and untrusted deployment environment

Application area

Telecommunications, Pervasive computing, banking applications, military applications, emergency networking, environmental purpose, ...

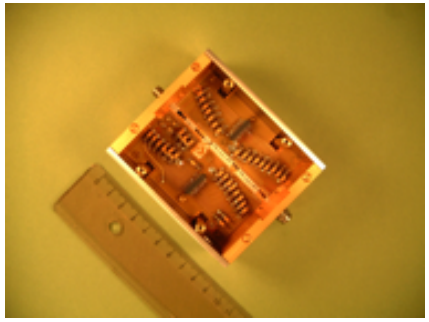
- Identification / Authentification
  - Banking Applications
  - Mobile phones (SIM cards)
  - Smartcard for WiFi
- Environment monitoring: Sensor networks using wireless data link transmissions and ad Hoc routing protocols
- And of course, ambient computing

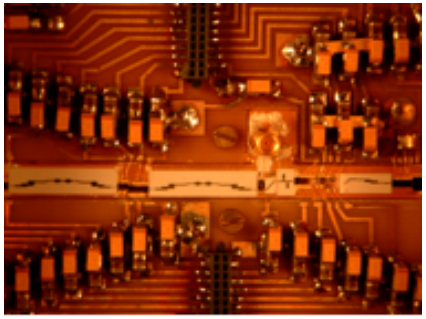
Hard/soft interface in wireless communications

In partnership with IEMN (Research institute in microelectronic area)

High rate communication interface for indoor communications

- 60 GHz UWB 100 Mb/s
  - Contention free
  - Directionnal antennas
- Use of a low rate (12 Mb/s) control channel
  - Random access (variation of CSMA/CA)
  - Omnidirectional reception
  - Directional emission
- Some challenges
  - Combination of packet scheduling and topology control
  - Beam switching and reduction of energy consumption





Objectives for the next four years

Focus on wireless sensor (and actuators) networks

- Go beyong simulations. . .

Objective 1 - Customization of evolving and communicating systems

- Dynamical customization
- Optimization of the communication stack

Objective 2 - Realistic wireless networking

- Position-based algorithms
- Hardware-software optimization

Safety is a transversal preoccupation

System and Networking for Portable Objects Proved to be Safe

Common project-team INRIA, Univ. Lille & CNRS

Thinking POPS as an usual target for general purpose software:

- Hide the complexity of the exotic hardware and communication management
- "Intelligence in system and framework instead of expertise of developers"
- Performance issue can be important
- Safety and security preoccupations are omnipresent

Relationships with companies

- Gemalto, SAP, Phillips, STm, Microsoft, Thales, Ericsson, Fiat
- Implied in the "Trade Industries cluster" in North of France

Outline

1. Presentation of POPS project-team
2. **Communications in sensor networks**
   - Overview of POPS activity in sensor networks
   - Example of key predistribution
3. Safety and security verifications in autonomous systems
   - General scheme for embedded verification
   - Applications/Experimental Results
4. Conclusion

Outline

1. Presentation of POPS project-team
2. **Communications in sensor networks**
   - Overview of POPS activity in sensor networks
   - Example of key predistribution
3. Safety and security verifications in autonomous systems
   - General scheme for embedded verification
   - Applications/Experimental Results
4. Conclusion

POPS in sensor networks

Applications

- European FP6 IST IP Project Wirelessly Accessible Sensor Populations (WASP) (2006-2009)
- RNRT Project Surpervise and Protect (SVP) (2006-2009)

Experimentations

- RNRT Project SensLab (2008-2011)
- European FP7 ICT IP Project Advanced Sensors and lightweight Programmable middleware for Innovative Rfid Enterprise applications (ASPIRE) (2008-2011)

Protocols and communication

- Energy efficient routing protocols
- Robustness and security

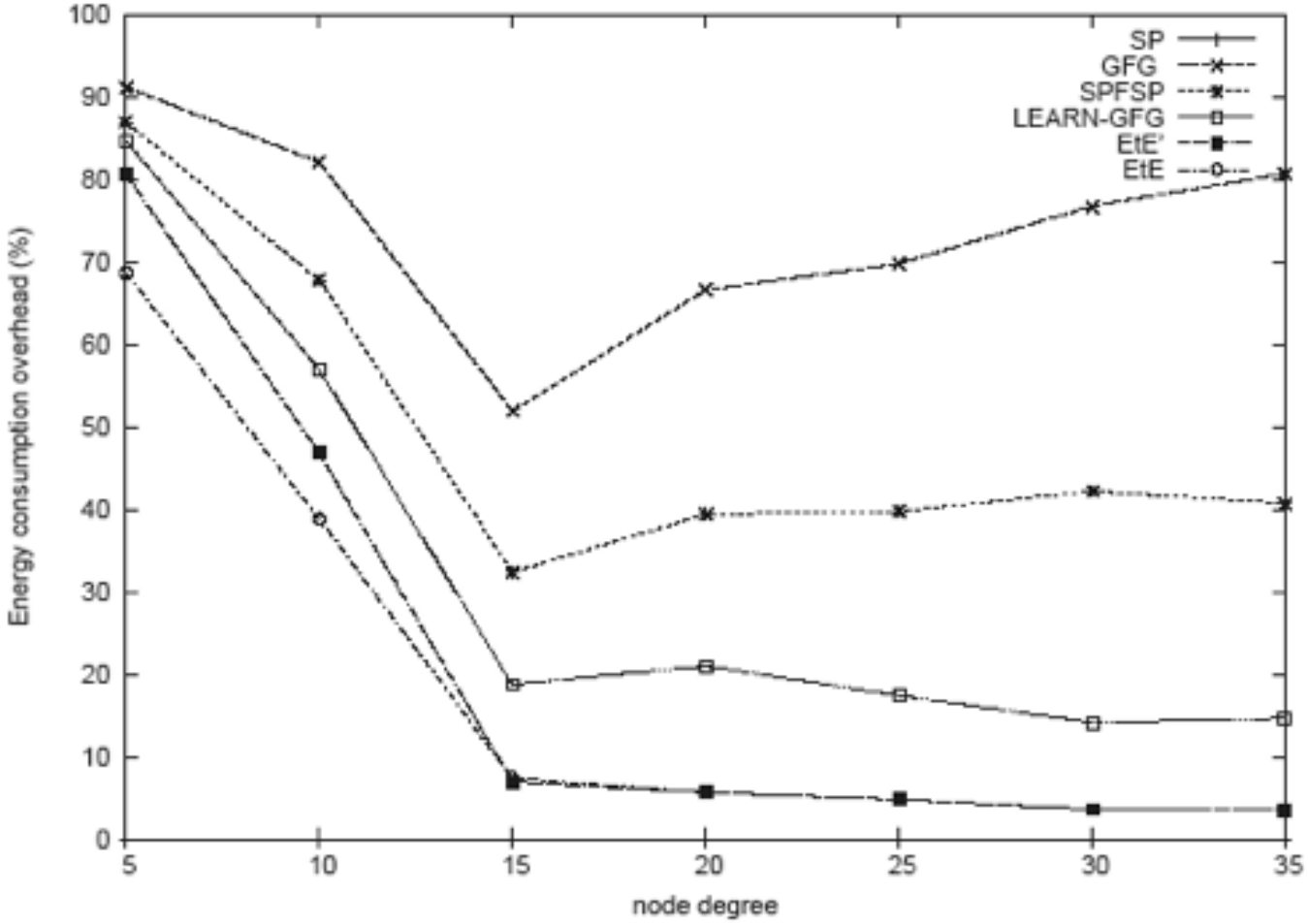
SensLab

Very large open wireless sensor network testbed

Position-based algorithms

With geographical information		
Basic algorithm	YES	(MFR)
Energy efficient (EE)	YES	(cost/progress,**ETE'**)
Garanteed delivery (GD)	YES	(FACE,GFG)
EE+GD	YES	(**ETE**)
Without geographical information		
Basic algorithm	YES	(VCap)
Energy efficient (EE)	YES	(**VCost**)
Garanteed delivery (GD)	YES	(**LTP**)
EE+GD	YES	(**current**)

Energy-efficient geographical routing



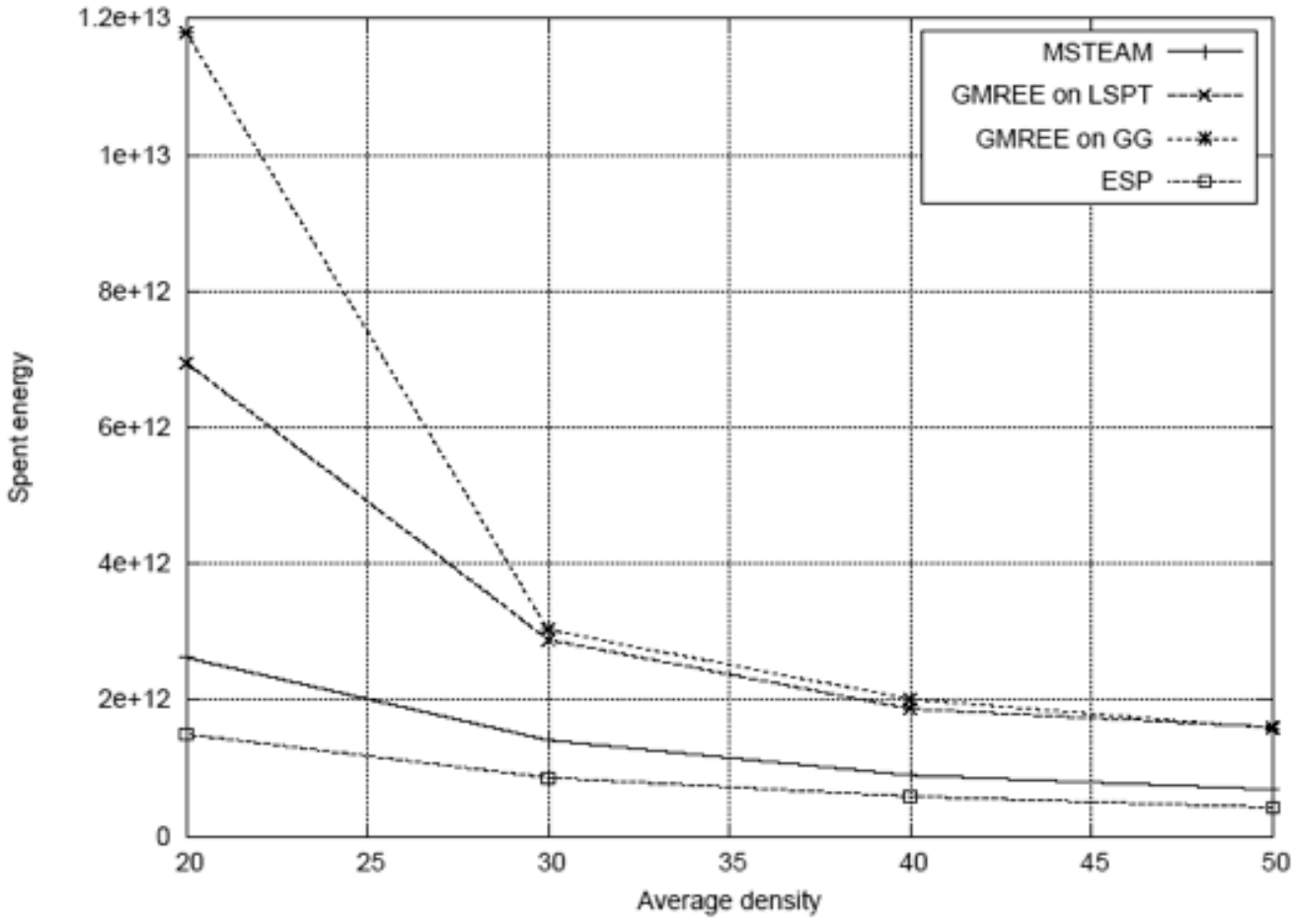
2005

2006

Beg. 2007

End 2007

Multicast routing MSTEAM performances



Ruiz et al. (2007)

MSTEAM
Centralized algorithm

Objectives

Proposing **energy-efficient** geographic routings with **guaranteed delivery** for **realistic** environments based on **INS** positioning

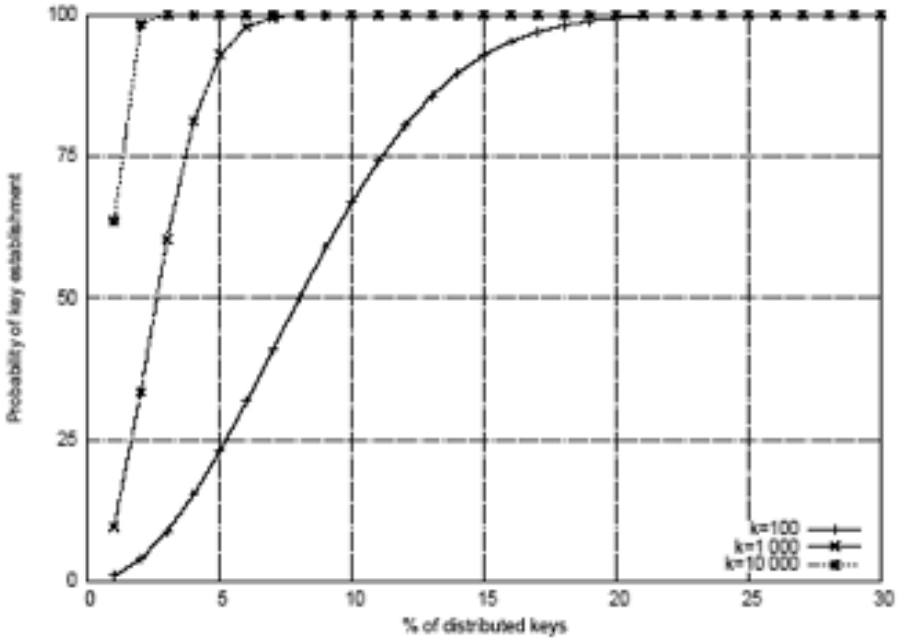
- Extending solutions to multicast routing and data collection
- Application and adaptation to RFID-based networks
- **INS** (Intertial Navigation Systems): hard to embed precise numerical integration algorithms, collaborative correction using neighbors, precise position in all environments
- Optimizations and corrections using data semantics and data correlation

Outline

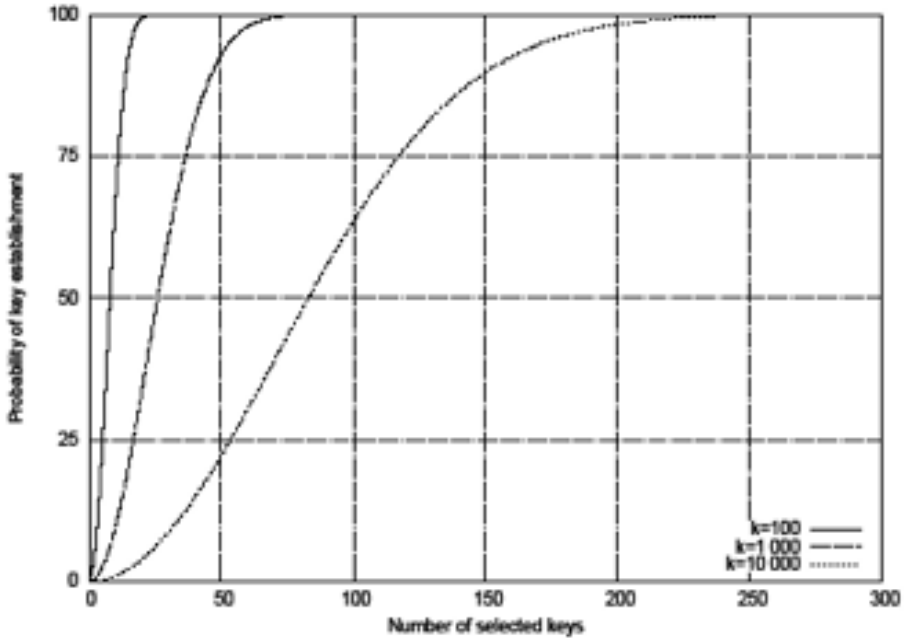
1. Presentation of POPS project-team
2. **Communications in sensor networks**
   - Overview of POPS activity in sensor networks
   - Example of key predistribution
3. Safety and security verifications in autonomous systems
   - General scheme for embedded verification
   - Applications/Experimental Results
4. Conclusion

Key predistribution in WSN

- Security issues in wireless sensor networks:
  - Fault insertion and pervasive listening
  - No collaboration problem
- Cryptography for communication between valid nodes
  - Public key mechanisms require too much computation problem
  - Use of symmetrical schemes
- Nodes can be captured
  - A single shared secret can be compromised by the capture of a single node
  - Each node contains a subset of a key space
    - ⋆ Let K be a set keys. Each node has a randomly chosen subset of K of size n
    - ⋆ Two nodes can communicate if they share a key
    - ⋆ Remark: if $2n > k$, two nodes can always communicate


Probability of key establishment
% of distributed keys
k=100
k=1 000
k=10 000

(a)

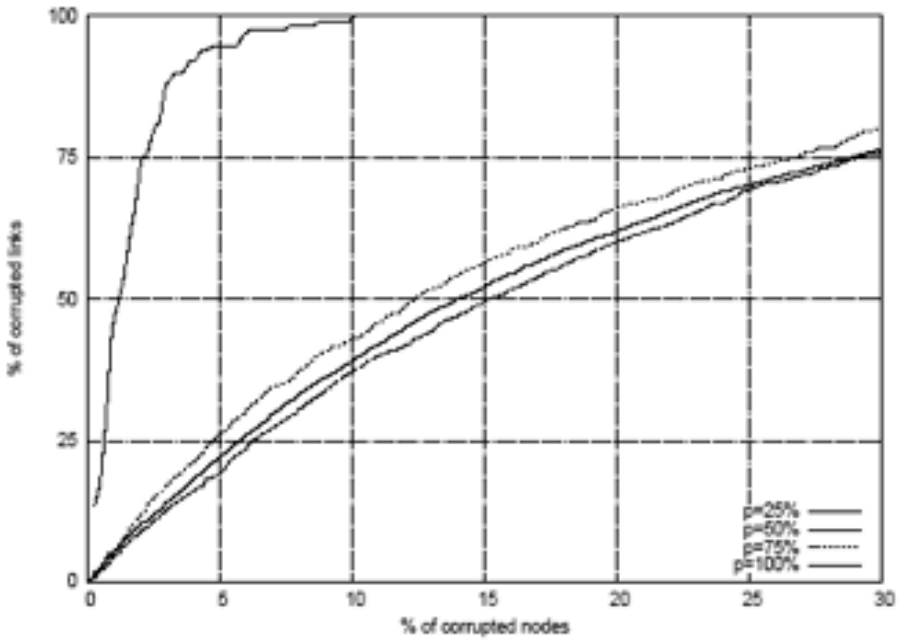

Probability of key establishment
Number of selected keys
k=100
k=1 000
k=10 000

(b)

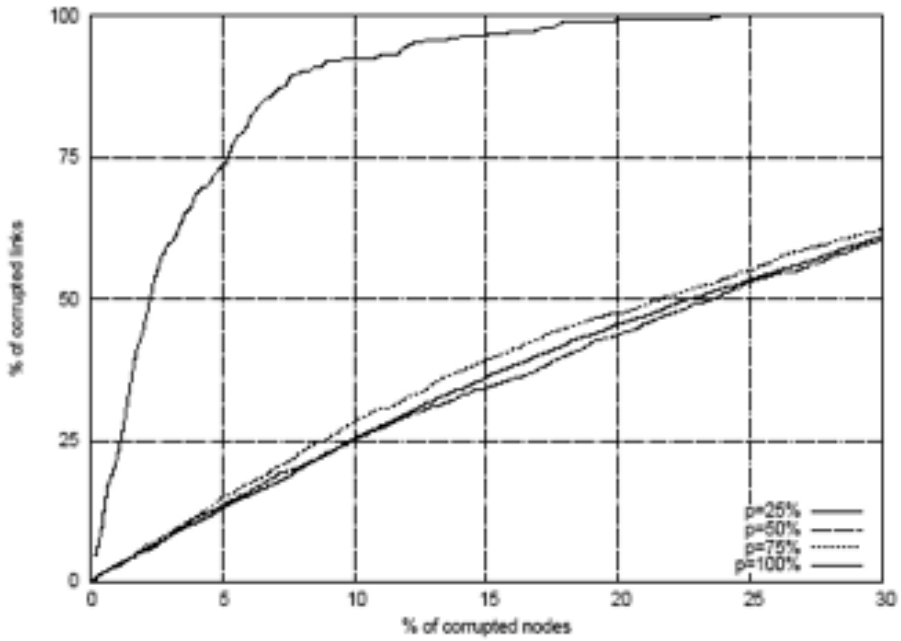
Contribution

- Connectivity evaluation based on unit disk graph
  - Two nodes can establish a key if and only if they are neighbors in the unit disk graph
  - Extension of Erdös-Renyi results
- Erasing keys
  - After key establishment - that includes neighborhood discovery - unused keys are erased
- Multi-deployed with non-disjoint key space and activity scheduling
  - Connectivity between different deployment is ensured via key sharing instead of bridge nodes
  - Deletion of keys with the objective of preservation of key set in neighborhood

Erasing of unused keys



(a) $n = 50$



(b) $n = 100$

Fig. 4. Percentage of corrupted links versus percentage of captured nodes with key erasing.

Outline

1. Presentation of POPS project-team
2. Communications in sensor networks
   - Overview of POPS activity in sensor networks
   - Example of key predistribution
3. **Safety and security verifications in autonomous systems**
   - General scheme for embedded verification
   - Applications/Experimental Results
4. Conclusion

Outline

1. Presentation of POPS project-team
2. Communications in sensor networks
   - Overview of POPS activity in sensor networks
   - Example of key predistribution
3. **Safety and security verifications in autonomous systems**
   - General scheme for embedded verification
   - Applications/Experimental Results
4. Conclusion

Goal

Intelligence in operating system and framework instead of expertise of developers

Main features

- System that is extensible, real-time **and** secured
- Code optimizations for target platforms that are verifiable
- Support for security of applications

Platforms

- CAMILLE ➠transfered to Gemalto
- JITS

Safety of small autonomous embedded systems

- Openness, mobility, post insuance, ...
- Need of static reasoning because of low performances
- Autonomicity because of potential hostile environments

Object-Oriented

- Virtual invocations ➠not possible to decide which code will be executed

Openness – Dynamic loading

- New sub-classes, new calling contexts ➠stated results on running systems must still hold
- Mobile code may come from hostile environment ➠mobile code must be "verified" on execution site

Object-Oriented + Open ➠Highly dynamic

Small and autonomous embedded systems

- Properties must be verifiable with few ressources

Compositional static analysis based on contracts

A contract is associated to each method

- Depending on the property
- Natural grain for object-oriented systems

Support for dynamic loading and openness: Compositionnality

- New code must respect required contracts

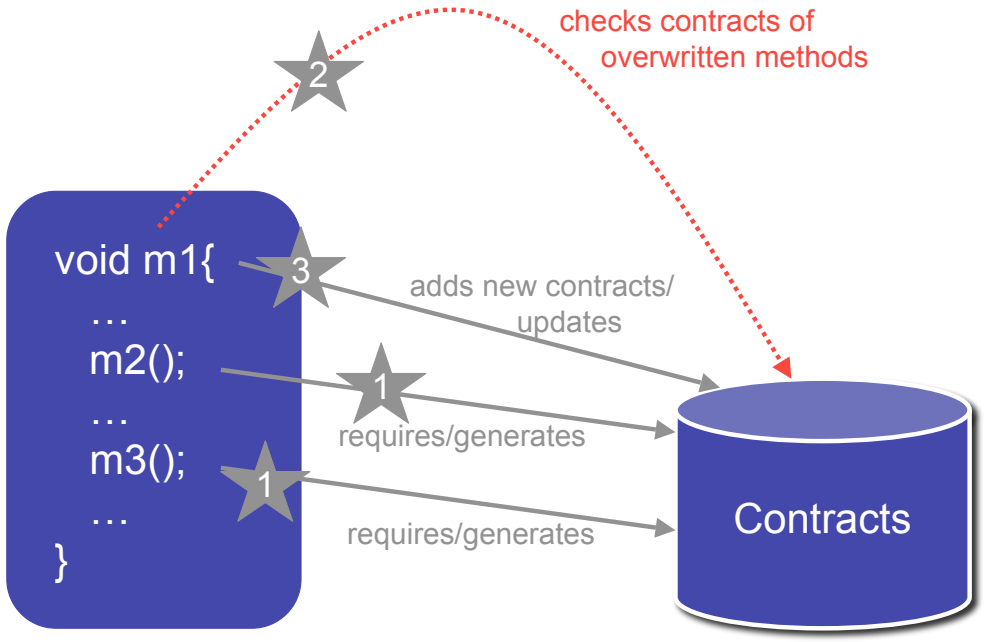
  ➠already established properties still hold
- New code uses contracts of old code

  ➠No need to re-analyse old code in new context

On small embedded systems

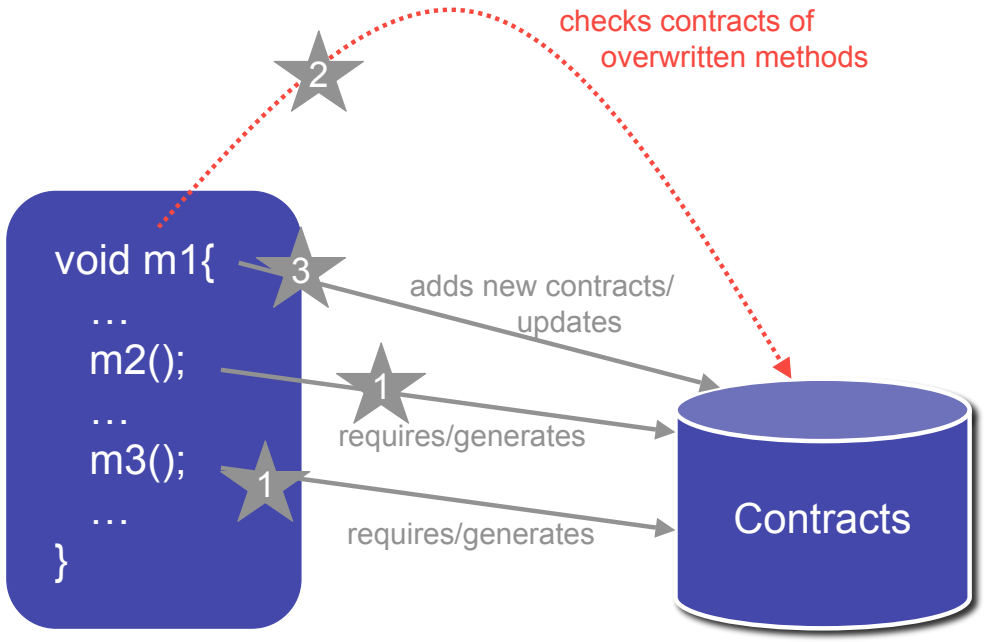
- Use of some light version of Pcc to verify method contract when loading the method code
- Possibility to verify a method when the called methods are not available

Use of contracts



$$\frac{(\mathcal{V}, u_n :: \cdots :: u_0 :: s, Mem, P) \qquad \mathcal{C}_m}{(\mathcal{V}, ret :: s, Mem', P \oplus \mathcal{C}_m)} \texttt{ invoke m}$$

Use of contracts



$$\frac{(\mathcal{V}, u_n :: \cdots :: u_0 :: s, Mem, P) \qquad \mathcal{C}_m}{(\mathcal{V}, ret :: s, Mem', P \oplus \mathcal{C}_m)} \texttt{ invoke m}$$

Inter-method analysis

Problems caused by openness

When analysing "`invoke` $\mathcal{M}'$" on o in $\mathcal{M}$

- The contract of $\mathcal{M}'$ may not be available (ex: $\mathcal{M} = \mathcal{M}'$)
- Exact type of o may not be known

Solution

- Analysis of a set of classes
  - Solve the recursivity problem
  - More flexible, add support for interfaces, abstracts classes, ...
- Use of approached contracts when the exact type is not known

Analysis of a group of classes

- Starts with all contracts at "bottom"
- Fix-point computation

Inter-method analysis

Problems caused by openness

When analysing "`invoke` $\mathcal{M}'$" on o in $\mathcal{M}$

- The contract of $\mathcal{M}'$ may not be available (ex: $\mathcal{M} = \mathcal{M}'$)
- Exact type of o may not be known

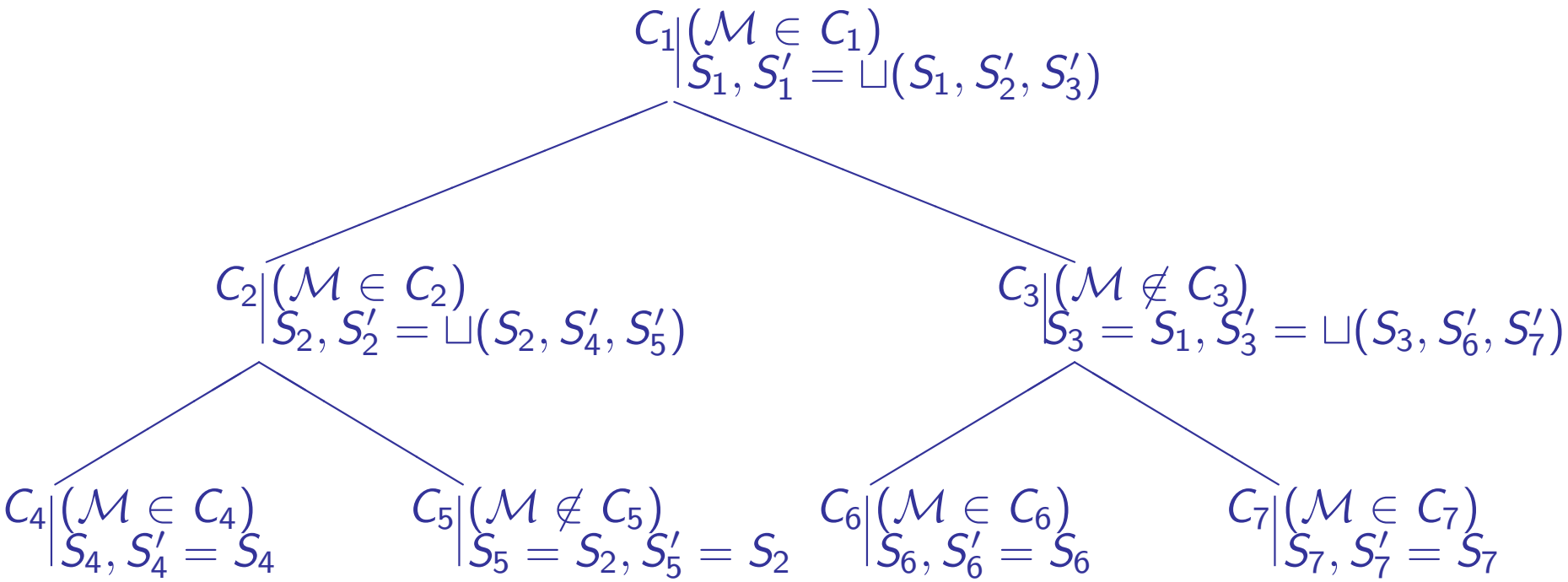
Solution

- Analysis of a set of classes
  - Solve the recursivity problem
  - More flexible, add support for interfaces, abstracts classes, ...
- Use of approached contracts when the exact type is not known

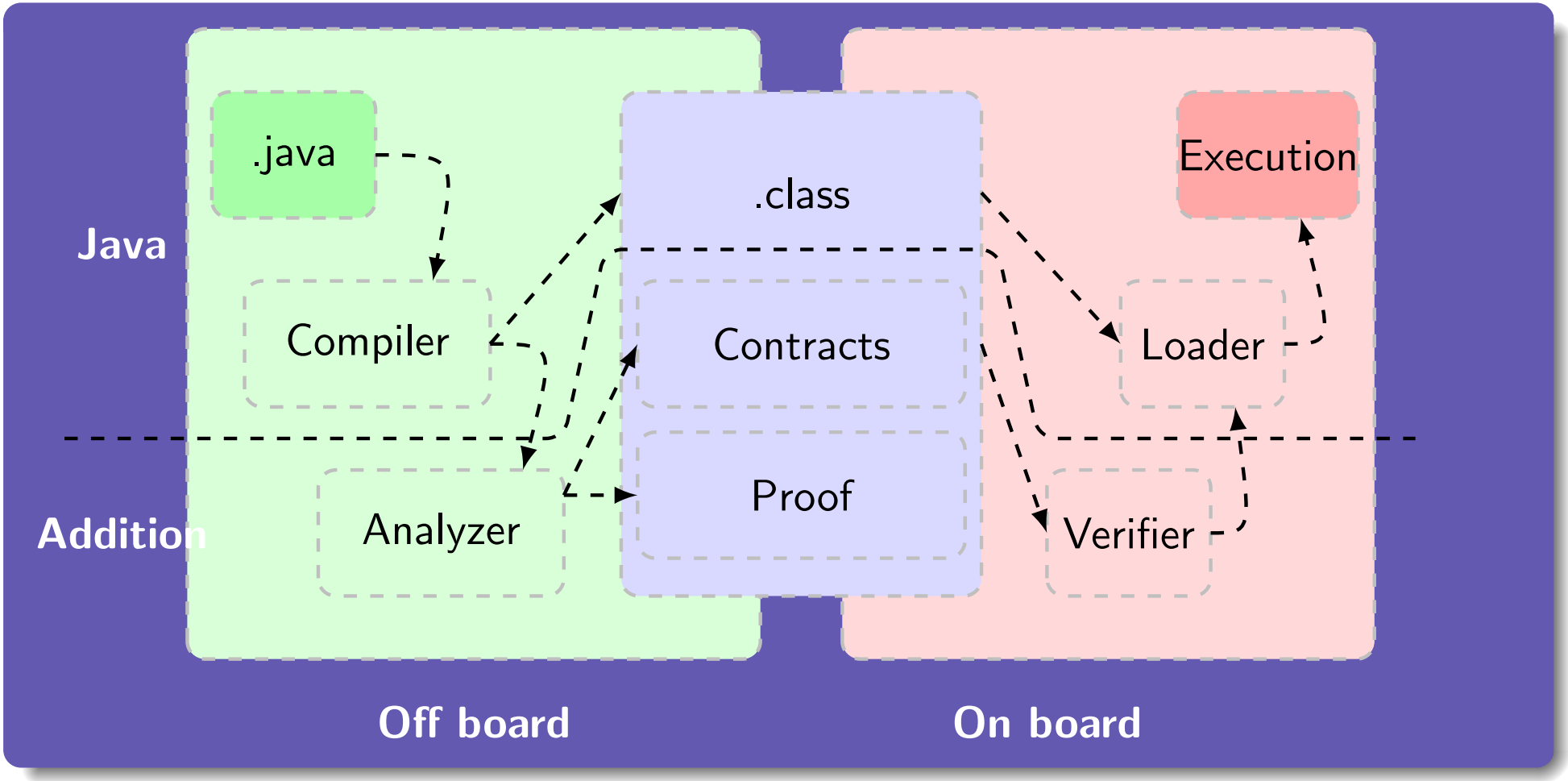
Analysis of a group of classes

- Starts with all contracts at "bottom"
- Fix-point computation

Approached signatures



Analysis scheme



Embedded verification

Ligthweight bytecode verification - Eva Rose

- Elegant Java bytecode verification in smart cards
- Idea: "*It is easier to verify a result than to establish it*"
- Two phases algorithm

Extension

- Extension of the technique using the contracts for all properties that have a lattice structure
- Contract repository of already loaded methods
- A method is loaded with:
  - Proof annotations
  - Necessary contracts
- Verification is then linear: a method is accepted if its contract can be verified and is coherent with the contract repository of the system

Outline

1. Presentation of POPS project-team
2. Communications in sensor networks
   - Overview of POPS activity in sensor networks
   - Example of key predistribution
3. **Safety and security verifications in autonomous systems**
   - General scheme for embedded verification
   - Applications/Experimental Results
4. Conclusion

Applications

- Escape analysis with extension for constructors and factories
- Information flow analysis
- WCET computation
- ...

Verification

- Addition of proof annotations and signatures used as attributes of the file `.class`
- Temporary signatures dictionnary
- The verifier is implemented in a class loader "SafeClassLoader"
- Various SCL and other Class Loaders may be used and the lookup mechanism for signatures is the same than the delegation mechanism for class loading

Results for IF

			Prover (external)					Verifier (embedded)				
Benchmark	Classes	Methods	Class iterations	Bytecode iterations	Analysis time (s)	Average memory (Kb)	Maximum memory (Kb)	Execution time CL (ms)	Execution time SCL (ms)	Verification time SCL (ms)	Average memory (Kb)	Maximum memory (Kb)
Dhrystone	5	21	3	1.47	5.4	4.26	35.94	121	473	402	0.78	3.80
fft	2	20	3	1.82	6.8	1.72	7.98	58	235	211	0.67	3.33
_201_ compress	12	43	3	2.21	7.7	3.52	20.84	321	599	364	0.85	4.31
_200_check	17	109	4	1.20	15.2	5.04	34.60	128	896	810	1.26	6.64
crypt	2	18	3	1.66	9.8	2.22	20.78	72	314	268	0.83	6.96
lufact	2	20	3	2.31	3.9	4.35	23.33	526	916	359	0.81	2.29
raytracer	12	72	5	1.85	8.7	2.54	25.23	80	587	544	0.84	3.33
Pacap	15	92	4	1.06	7.5	6.62	92.84	30	402	385	1.01	6.01

Results for IF

Benchmark	Initial class size (Kb)	Annotated class (Kb)	Signatures (%)	Labels proof (%)	External methods (%)	External fields (%)
Dhrystone	8.2	14.4	8.00	52.22	5.15	0.20
fft	6.8	15.1	16.60	91.09	7.23	0
_201_ compress	20.1	28.3	3.95	23.66	4.36	0.34
_200_check	46.3	97.7	12.27	85.05	6.75	0.04
crypt	7.0	17.0	12.27	118.28	5.90	0.07
lufact	9.3	17.0	8.44	64.31	4.07	0.39
raytracer	24.0	42.8	20.44	36.12	12.06	0.57
Pacap	26.8	52.0	18.36	55.72	9.03	0.37

Outline

1. Presentation of POPS project-team
2. Communications in sensor networks
   - Overview of POPS activity in sensor networks
   - Example of key predistribution
3. Safety and security verifications in autonomous systems
   - General scheme for embedded verification
   - Applications/Experimental Results
4. **Conclusion**

Conclusion

- Short overview
- Ask more if you are interested...