Extending the analysis

Per estendere l'analisi vediamo qualche possibilità: la prima è quella di andare a vedere un po più da vicino le azioni quindi cosa fa l' attuatore? Fino a ora abbiamo soprattutto controllato che cosa fanno i sensori, sostanzialmente adesso vediamo di tracciare meglio le azioni dell' attuatore, per farlo ho bisogno però di estendere la mia analisi perché, come abbiamo detto l'analisi mi prevede il comportamento rispetto alle cose che in quel momento mi sembra interessante, quindi fino a ora non abbiamo controllato gli attuatori, adesso lo facciamo, quindi mi serve un'altra componente che chiamo alfa e che aggiunge e quindi si aggiunge alla tripla che avevamo prima quindi vado a controllare le azioni che si possono fare quindi in particolare vado a vedere quali azioni un nodo può scatenare.

The goal

Tracking actions a node may trigger on actuators

How

A new component in the analysis estimate

- Action collection α
  - Set of actions a node may trigger
- The analysis specification is consequently updated (e.g. $(\hat{\Sigma}, \kappa, \Theta, \alpha) \models_\ell B$)

A new rule for actuators

$$\frac{\gamma \in \alpha_\ell(j) \wedge (\hat{\Sigma}, \kappa, \alpha, \Theta) \models_\ell P}{(\hat{\Sigma}, \kappa, \alpha, \Theta) \models_\ell \langle j, \gamma \rangle.P}$$

- γ is an action of the actuator j
- γ is recorded inside α

In particolare, quindi, ho bisogno di una nuova regola per gli attuatori è una possibilità, naturalmente questo tipo di estensione si può fare anche in maniera diversa, però quello che posso fare è che ogni volta che analizzo un processo e vedo che il processo chiede all' attuatore j di fare l'azione gamma, vado a ricordarmi che gamma è una delle possibili azioni che j può fare nel nodo l e questo lo faccio controllando che gamma appartenga ad alfa l di j.

Formal results

Standard results (as before)

Subject reduction the analysis is sound w.r.t. the semantics

Existence of solution valid estimates form a Moore family

New result: action tracking α

1. correctly approximates the actions nodes may trigger
2. suffices to understand if an actuator is useless
3. basis for predicting the impact over the environment

IoT security

- An open problem and very difficult challenge
  - HP: Internet of Things Research Study. 2014
  - The Internet of (Broken?) Things
  - The Internet of Unpatched Things
  - Internet of Things (IoT) as Interconnection of Threats (IoT)
- Our goal is to use the analysis result to check (without re-computation)
  - If a sensible piece of data passes in clear
  - If nodes communicate according to some policies

Introducing cryptographic primitives

Processes

$P ::= \ldots$ IoT-LySA constructs as above

decrypt E as $\{E_1, \cdots, E_j; x_{j+1}, \cdots, x_k\}_{k_0}$ in P

Terms

$E ::= \ldots$ IoT-LySA terms

$\{E_1, \cdots, E_n\}_{k_0}$ encrypted data

L'altra cosa da fare è introdurre, se vogliamo parlare di sicurezz e secrecy, le primitive crittografiche. Ora non ho bisogno di spiegare molto qui perché in realtà l' ereditiamo direttamente dal LySa e quindi abbiamo nei processi un modo per fare decryption sempre con pattern matching per semplificarci la vita...

...e poi abbiamo naturalmente la possibilità di inserire encryption tra le possibili espressioni che possiamo scambiare.
Naturalmente supponiamo che ci siano già delle chiavi fissate, quindi come vedete, rispetto al LySa abbiamo un valore fissato k e un'espressione che può variare, quindi supponiamo per semplicità che le chiavi sono un certo numero di finito e che siano conosciute sin da prima.

Introducing cryptographic primitives

Processes

$$P ::= \ldots \qquad \text{IoT-LySA constructs as above}$$
$$\text{decrypt } E \text{ as } \{E_1, \cdots, E_j; x_{j+1}, \cdots, x_k\}_{k_0} \text{ in } P$$

Terms

$$E ::= \ldots \qquad \text{IoT-LySA terms}$$
$$\{E_1, \cdots, E_n\}_{k_0} \qquad \text{encrypted data}$$

- Nodes, sensors, actuators unchanged
- Keys are finitely many and are known in advance (à la ZigBee)

Semantics of decryption

(Decr)

$$\frac{[\![E]\!]_\Sigma = \{v_1, \cdots, v_k\}_{k_0} \ \wedge \ \bigwedge_{i=1}^{j} v_i = [\![E_i]\!]_\Sigma}{\Sigma \parallel \text{decrypt } E \text{ as } \{E_1, \cdots, E_j; x_{j+1}, \cdots, x_k\}_{k_0} \text{in } P \rightarrow \Sigma[v_{j+1}/x_{j+1}, \cdots, v_k/x_k] \parallel P \parallel B}$$

decryption (the same key) + matching

a questo punto la semantica della decryption è totalmente uguale a quella che abbiamo visto per il pi calcolo, sostanzialmente la decryption semanticamente la posso fare quando la tupla che vado a decrittare, questa volta con la chiave, e quella coincide per forza, è della stessa arità e i primi j elementi sono quelli che chi fa la decryption si aspetta, e a questo punto ho tutti gli ingredienti per stabilire delle proprietà di sicurezza, guarda caso, tanto per ritornare su cose che abbiamo visto possiamo stabilire delle proprietà analoghe a quelle che abbiamo visto per il pi calcolo, quindi possiamo andare a vedere le proprietà di secrecy.

Preventing leakage

Assumptions

- Values $=$ Public $\uplus$ Secret
- A drop of secret makes a term secret
- Encryption makes a secret public

Using κ to detect if a secret passes in clear

If $(\ell, \langle\!\langle v_1, \ldots, v_s, \ldots, v_n \rangle\!\rangle) \in \kappa(\ell')$ and v_s is secret $\Longrightarrow$
possible problem: at runtime the tuple can flow in clear from ℓ to ℓ'

Constraining communications

Assumptions

A policy $\phi(\ell, \ell')$ governs communications from ℓ to ℓ'
(e.g. no read up-no write down policy)

Using κ to detect policy violations

If $(\ell, \langle\langle v_1, \ldots, v_n \rangle\rangle) \in \kappa(\ell')$ and $\phi(\ell, \ell') = \mathit{false} \implies$
possible problem: at runtime a violation may occur

L'altra possibilità invece rispetto alla comunicazione, e alle traiettorie della comunicazione è quella della policy come dicevamo, quindi io posso decidere e lo lascio apposta parametrico, posso decidere una qualsiasi policy che mi dice qual è il tipo di comunicazione ammissibili tra due nodi l e l' che può essere ad esempio non read up e no write down ma può essere una qualsiasi policy che è legata alla forma particolare del sistema che stiamo analizzando, quindi posso decidere indipendentemente da chi sta sopra o chi sta sotto, che non ci devono essere comunicazione tra il nodo l1 e il nodo l3 per motivi che dipendono dal design di quel sistema. Ecco allora potete immaginare che anche questo tipo di proprietà possono essere catturabili dal nostro framework perché vado a controllare la componente di nuovo k e laddove vedo che c'è una possibile comunicazione tra l e l' vado a controllare se la policy che ho mi consente questo tipo di comunicazione e quindi decidere se c'è una violazione possibile a run time

Further developments

- Tracking sensitive and untrustworthy data in IoT with a *taint analysis* of IoT-LySA
- Tracking data trajectories and provenance of data in IoT with IoT-LySA and with Klaim (language designed for specifying the behaviour of distributed and coordinated processes at a suitable level of abstraction)

A questo punto smetto di farvi vedere questi lucidi perché passo ad un'altra possibile estensione che è sotto la forma di analisi di tipo teint e questo lo faccio con un'altro blocco di lucidi e quindi vediamo qualcosa e che avete già visto in un'altra forma e quindi dovrebbe esservi familiare.