Présentation de VALIDY TECHNOLOGY

Technologies, Applications, Marchés

IT AND ECONOMIC SECURITY

www.validy.com

Génèse 1/2

Une vision :

Que l'informatique ne pourra continuer son déploiement au plan mondial sans que les problèmes de sécurité ne soient pris sérieusement en compte.

Dès 1991 des travaux exploratoires sur la sécurité informatique ont été menés (identification biométrique, login par carte à puce,…)

Une équipe pluridisciplinaire (électroniques, informatiques, réseaux), sans apriori et novice sur bien des points en sécurité informatique. Une équipe ouverte, prête à explorer des nouveaux champs des possible.

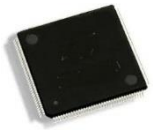
En 1998, une idée : Protéger des programmes en soustrayant de la vue des pirates une partie de l'état du programme en cours d'exécution.

VALIDY
IT AND ECONOMIC SECURITY
www.validy.com

Génèse 2/2

Un model de protection soustractif :

(Versus model additif)



- Utilisation d'un composant « hardware » pour cacher une partie du programme

- Pour être efficace, la partie cachée doit respecter 4 critères :

  - Doit être essentielle au programme
  - Sa communication ne doit pas être rejouable
  - Doit être non simulable
  - Ne doit pas ralentir l'exécution du programme

IT AND ECONOMIC SECURITY

www.validy.com

Rupture technologique

Un changement de paradigme :

- Un logiciel protégé par VALIDY TECHNOLOGY **se défend lui même** contre les attaques.
- La protection ne requiert et ne dépend d'aucune tierce partie ou autorité.
- La notion même d'extérieur et d'intérieur n'a pas de sens pour VALIDY TECHNOLOGY.
- VALIDY TECHNOLOGY est efficace y compris contre les nouvelles attaques qui n'ont par nature pas été répertoriées. (**Zero Day Attack)**

IT AND ECONOMIC SECURITY

www.validy.com

Rupture technologique

- Un logiciel protégé peut être mise à jour en toute sécurité par l'éditeur de logiciel.

- Validy Technology est aussi une solution réelle et efficace contre les menaces avancées permanentes (**Advanced Persistent Threats - APT)**

- Validy fournit un espace de confiance durant tout le cycle de vie d'une donnée.

    - Acquisition
    - Stockage
    - Traitement
    - Restitution

IT AND ECONOMIC SECURITY

www.validy.com

Capacités de Validy Technology

VALIDY Technology est un “jeton” de sécurité connecté au processeur principal. Il peut se matérialiser sous la forme d’un composant électronique ou être **directement intégré au processeur principal**.

Un compilateur automatise la protection du logiciel à protéger. La protection est unique et sous le seul contrôle du fabricant .

IT AND ECONOMIC SECURITY

www.validy.com

Fonctionnement

Conserver certaines variables du programme à l'intérieur du jeton.

- Modifier, de manière sécurisée, ces variables à l'intérieur du jeton.
- Sortir l'information du jeton basée sur ces variables uniquement lorsque le programme en a réellement besoin

Le jeton implémente un "co-processeur"

- Reçoit ses instructions encryptées.
- Décrypte les instructions puis les exécute.
- Stocke les variables du programme dans des registres ou dans de la mémoire propre.
- Détecte les attaques.

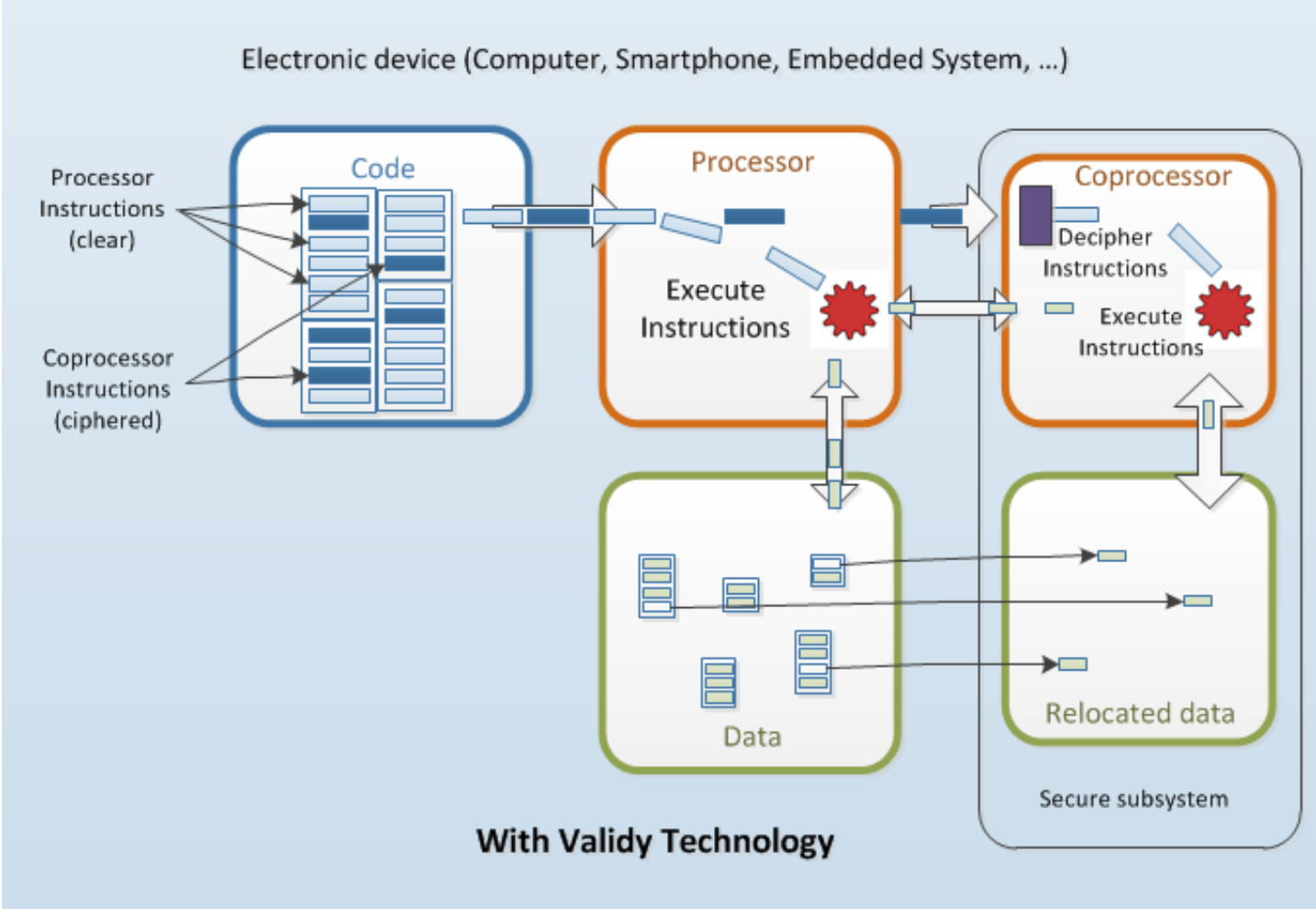
En cas d'attaque le jeton peut s'arrêter de fonctionner

- Dans ce cas le programme stoppe.

IT AND ECONOMIC SECURITY

www.validy.com

Fonctionnement



VALIDY
IT AND ECONOMIC SECURITY
www.validy.com

Modèle de programmation

- Le jeton peut mettre en œuvre toutes les ressources standards requises par les environnements de programmation modernes:
  - registres
  - piles
  - tas
- La pile et une partie des registres peuvent être thread spécifique
- Les ressources peuvent être virtualisées à condition que les données soient cryptées par une clé de session avant d'être paginées.

Le jeton

- Le jeton peut être
  - un microcontrôleur, programmé pour se comporter comme un coprocesseur
  - un circuit spécifique, créé spécialement pour cet usage
- Il met en œuvre des registres
- La plupart des instructions utilisent trois registres: 2 source et un de destination
- Le jeu d'instructions utilise des «Tags» pour détecter les attaques

Exemple de chaine d'instructions :

```
LDI     R1<tag1> 0
ADD     R3<tag3> R1<tag1> R2<tag2>
```

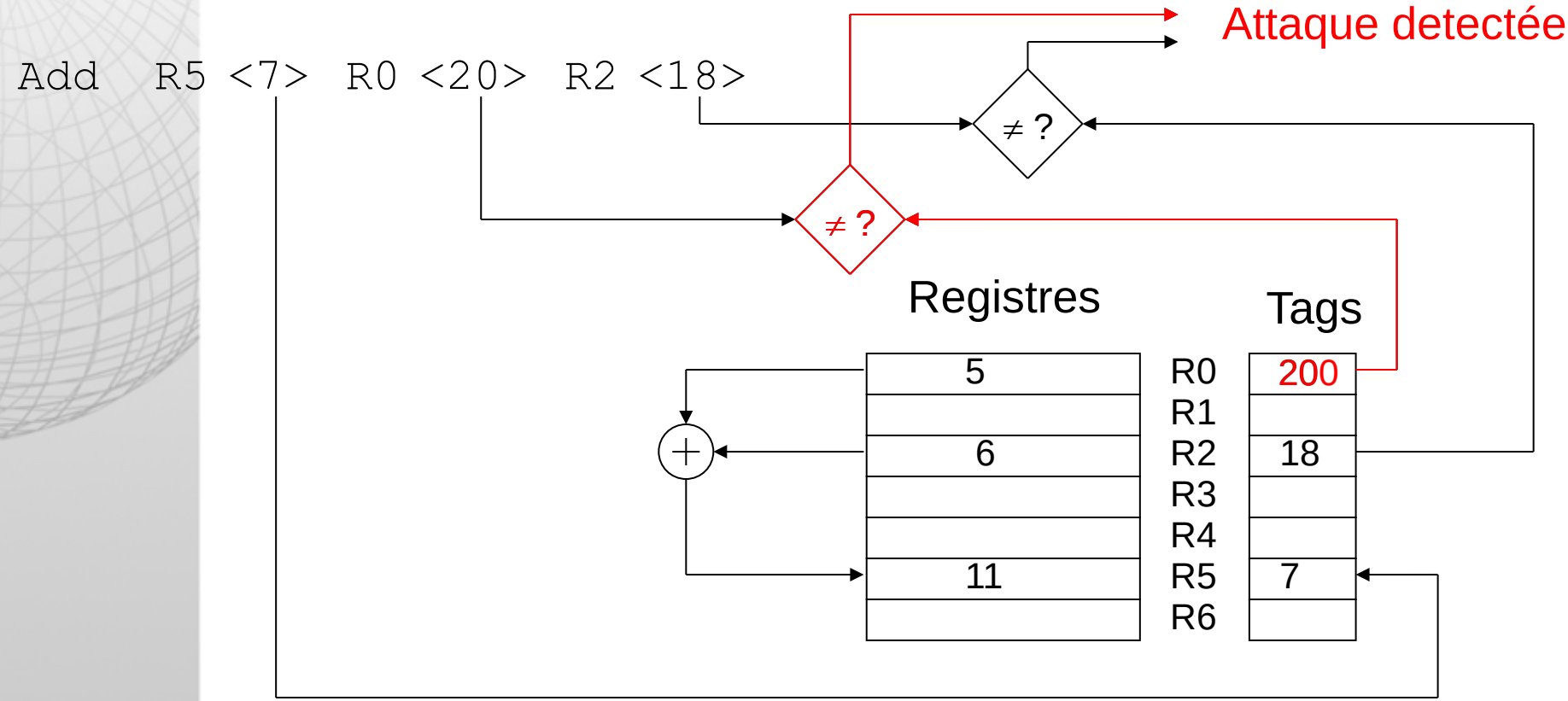
- Instructions pour "fusionner" les tags

```
LDI R1<tag1> 0
…
TGMV  R2<tmpTg>  R1<tag1> <tag2>
ADD     R1<tag2> R2<tmpTg> R3<tagR3>
```

IT AND ECONOMIC SECURITY

www.validy.com

Détection des attaques

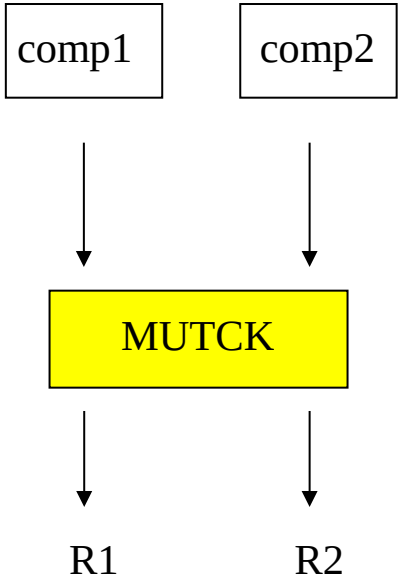


Verification de l'enchainement des instructions

Protection mutuelle

- Il est possible de lier l'exécution réussie d'un calcul à l'exécution réussie d'un autre calcul

R1<tag1> ← comp1
R2<tag2> ← comp2
XOR R3<tagTmp> R2<tag2> R2<tag2>
ADD R1<newTag1> R1<tag1> R3<tagTmp>



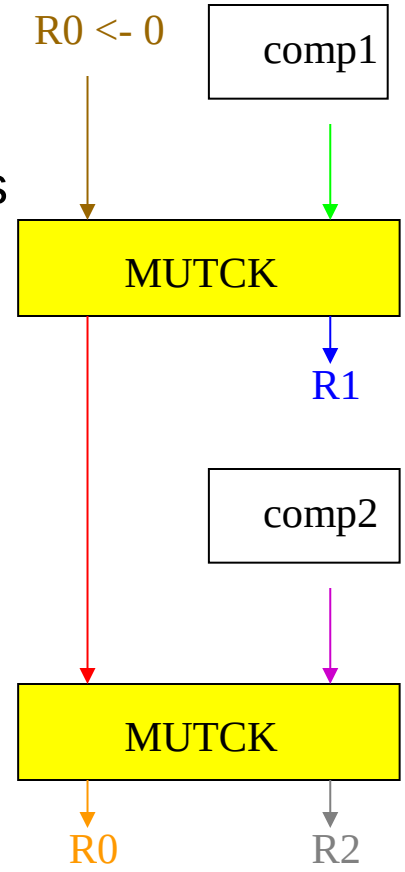
- Ce mécanisme peut également protéger mutuellement deux (ou plusieurs) calculs

MUTCK R1<newTag1> <tag1> R2<newTag2> <tag2>

Registre de lien et Protection série

- Un des "calculs" peut initialiser un registre avec une valeur quelconque
- Puis ensuite utiliser ce registre pour "relier" tous les autres calculs:

```
LDI        R0 <t0> 0
R1 <t1>    <-  comp1
MUTCK      R0 <t10> <t0> R1 <t11> <t1>
R2 <t2>    <-  comp2
MUTCK      R0 <t20> <t10> R2 <t12> <t2>
```



Verrouillage des calculs additifs

- Supposons que des calculs comp1, ... COMPn soient protégés en série.
- Si COMPn est un calcul obligatoire alors comp1, ... COMPn-1 vont devenir également obligatoire

IDEAL pour des vérifications d'intégrité!

IT AND ECONOMIC SECURITY

www.validy.com

Protection du graphe d'appel

- Chaque fonction peut maintenir un registre de lien et protéger ainsi tous les calculs effectués dans le jeton (En utilisant des instructions "join" si nécessaire)
- Une fonction appelante peut vérifier l'exécution d'une fonction appelée par le passage d'un registre de lien à la fonction appelée et en vérifiant après retour le tag du registre de lien
- Une fonction appelée peut vérifier l'identité de sa fonction appelante en vérifiant le tag de son registre de lien

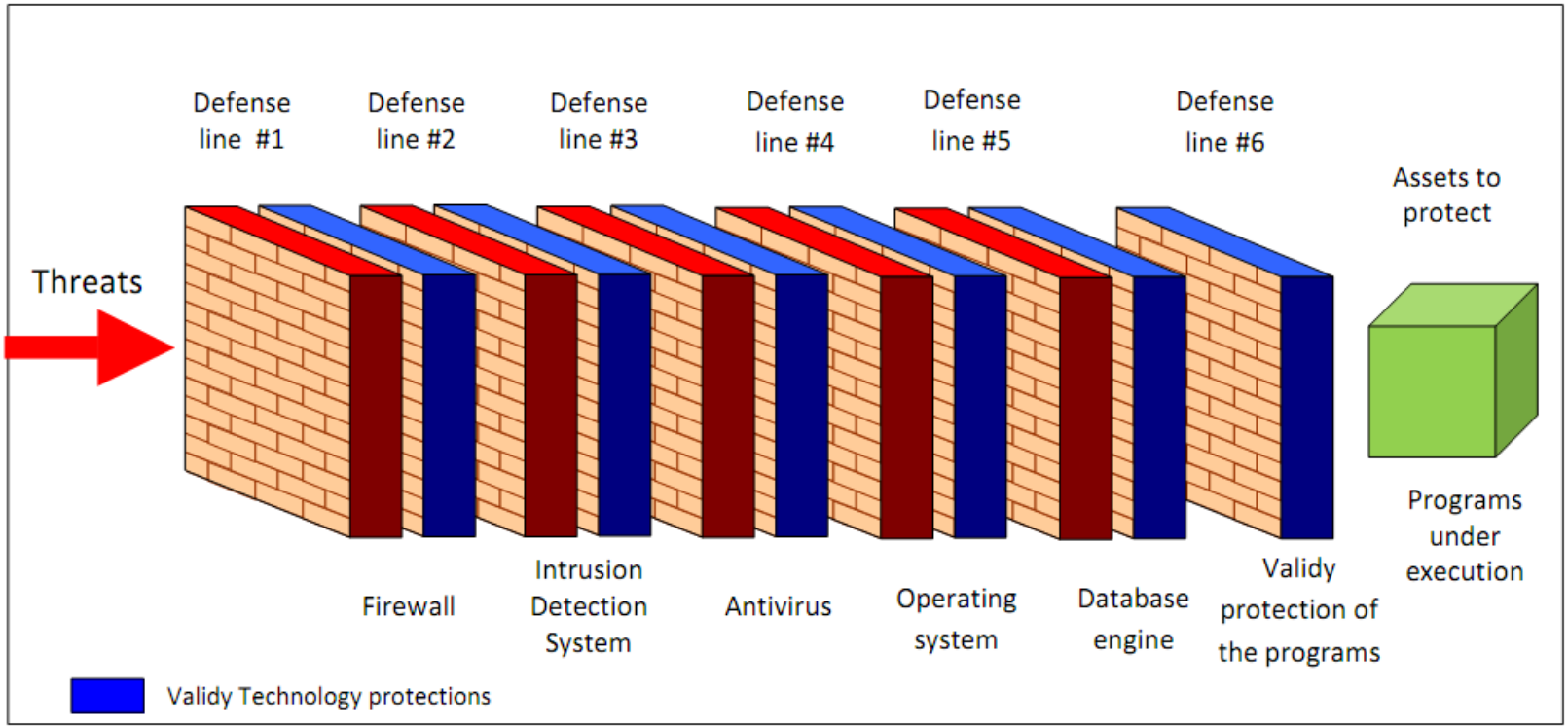
Protection mutuelle des programmes

- Si deux programmes utilisant des jetons communiquent, ils peuvent mutuellement se protéger:
  En utilisant le dialogue, chaque programme envoie périodiquement un challenge cryptographique à l'autre programme et arrête de travailler si la réponse est incorrecte
- Un serveur peut protéger ses clients
- Un serveur de mise à jour en temps réel peut protéger des applications à distance
- …

VALIDY
IT AND ECONOMIC SECURITY
www.validy.com

Defense en profondeur avec Validy Technology

Example of "Numerical Self Defense In Depth"



The above figure symbolically represents the security obtained by adding the Validy Technology protection to each traditional defense layer, but also to the Operating System, the Database engine and to the programs themselves.

IT AND ECONOMIC SECURITY

www.validy.com

Marchés / Bénéfices clefs de la technologie

VALIDY Technology permet aux fabricants de matériel électronique d'être sûr que leurs produits fonctionneront toujours conformément à leurs spécifications.

Leurs produits ne peuvent en aucune façon être corrompus et détournés de leurs fonctionnement normal.

La protection fournie par VALIDY est réellement efficace et sans équivalent dans le monde.

VALIDY
IT AND ECONOMIC SECURITY
www.validy.com

Applications/Marchés

- Une technologie difficilement classable, basique et générique
- Dont les domaines d'applications sont aussi nombreux que transverses
- Qui adresse des marchés aussi divers que ceux de l'électronique et l'informatique au plan mondial.

Exemples :

- **Protection des infrastructures d'Importance Vitale** (réseau de distribution d'électricité, réseau ferroviaire, réseau de distribution d'eau, réseau de télécommunication, réseau bancaire…)
- **Protection des systemes embarqués** (Téléphone portable, voiture, terminaux de paiement par carte bancaire, routeurs,…)

IT AND ECONOMIC SECURITY

www.validy.com

Applications/Marchés

Validy Technology peut être utilisé à la place d'un **TPM (Trusted Plateform Module**) en étant plus simple dans sa mise en œuvre.

Validy technology est aussi un **antivirus** ne nécessitant pas de connaitre la signature d'un virus.

VALIDY Technology est idéal pour protéger les **systèmes SCADA** (Supervisory Control And Data Acquisition) y compris contre les attaques Zero Day.

VALIDY Technology peut être utilisé comme un **outil de DRM** (Digital Right Management).

VALIDY Technology est aussi un système d'**Information Assurance**.

VALIDY Technology est aussi une solution pour sécuriser les **systèmes embarqué**s.

VALIDY Technology est une solution idéale pour sécuriser **l'Internet des Objets (IoT)** .

IT AND ECONOMIC SECURITY

www.validy.com

Protection d'un réseau de distribution d'énergie

Validy Technology est une technologie particulièrement pertinente pour protéger par exemple un réseau de distribution d'énergie (SCADA).

Capable de protéger le centre de commandement et contrôle

Ainsi que tous les équipements répartis sur le territoire qui assurent la distribution d'énergie.

VALIDY
IT AND ECONOMIC SECURITY
www.validy.com

Validy Technology et l'Internet des Objets

- Validy technology est une solution idéale pour protéger tout objet connecté à Internet. Les prévisions sont de 80 Milliards d'objets connectés à Internet d'ici 2020.

- Pour rappel Validy Technology peut être intégré par exemple dans le processeur QUARK d'INTEL ou dans la famille de processeurs d'ARM, ST MICROELECTRONICS … ou tout autre microprocesseur.

VALIDY
IT AND ECONOMIC SECURITY
www.validy.com

Propriété Intellectuelle

Validy c'est deux structures : Validy Net Inc (Oregon, USA) et Validy SA (Drôme, France).

Validy, c'est 10 Millions d'euros investis principalement dans la R&D (10 années), la recherche du modèle d'affaire, dont 2 millions d'euros dans la propriété intellectuelle :

- Un portefeuille de brevets composés de 10 brevets :

  - Brevet "Variable"
  - Brevet "Fonctions élémentaires"
  - Brevet "Renommage"
  - Brevet "Branchement conditionnel"
  - Brevet "Dissociation temporelle"
  - Brevet "Détection et coercition"

  - Brevet "Protection des bibliothèques"
  - Brevet "Protection des données d'échanges"
  - Brevet "Quatre groupes"
  - Brevet "Bouquet"

IT AND ECONOMIC SECURITY

www.validy.com

Contacts

Gilles SGRO – gilles.sgro@validy.com T 06 84 60 00 96

Jean-Christophe CUENOD – jcc@validy.com T 04 75 717 717

Questions - Réponses

IT AND ECONOMIC SECURITY

www.validy.com

Backup slides

Séminaire CAP'TRONIC - ESISAR du 19 Juin 2014

IT AND ECONOMIC SECURITY

www.validy.com

“Validy Technology” Presentation

Synthesis document, which due to its summary nature, necessarily contains approximations and imprecisions

IT AND ECONOMIC SECURITY

www.validy.com

1 Software protection

- To prevent someone from running a piece of software if a valid license has not been acquired for it.

  => prevents illegal copies

- To prevent anybody to tamper with a piece of software

  => insures integrity

IT AND ECONOMIC SECURITY

www.validy.com

1.1 Software protection versus Audio or Video protection

- The 2 problems are fundamentally different because:
  - Software never executes twice alike : Software is « Non replayable »
  - Audio or Video is always played in the same manner.

- A software protection mechanism can take into account the « non replayable » property whereas an Audio or Video protection mechanism cannot.

IT AND ECONOMIC SECURITY

www.validy.com

1.2 The « pirates »

- « pirates » always attack the weakest point of a protection
- « pirates » have very powerful tools
  - Software : debuggers, disassemblers, virtual machines, …
  - Hardware : scopes, logic analyzers, custom tools, …
- « pirates » can be very patient and can team-up to obtain a result.

1.3 The additive protection model

- The protection is obtained by adding some « complex » functions to a program (using cryptography, obfuscation, …)

- Since those protections are added to the program functionality, with enough efforts, a « pirate » is able remove them and reconstruct a completely functional unprotected program.

1.4 The subtractive protection model

- Use a secure hardware token to « hide » part of the program
- To be effective the hidden part must respect 4 criteria:
  - It must be essential to the program
  - Communication with it must be non replayable
  - It must be non simulable
  - It must not slow down the execution

2 Validy Technology inventions

- « Keep mandatory program state in a secure hardware token and modify it there. Output information based on the token internal state, only when needed by the program. »
- Based on 6 intertwined principles:
  - Variable
  - Temporal dissociation
  - Elementary functions
  - Renaming
  - Detection and coercion
  - Conditional branches

IT AND ECONOMIC SECURITY

www.validy.com

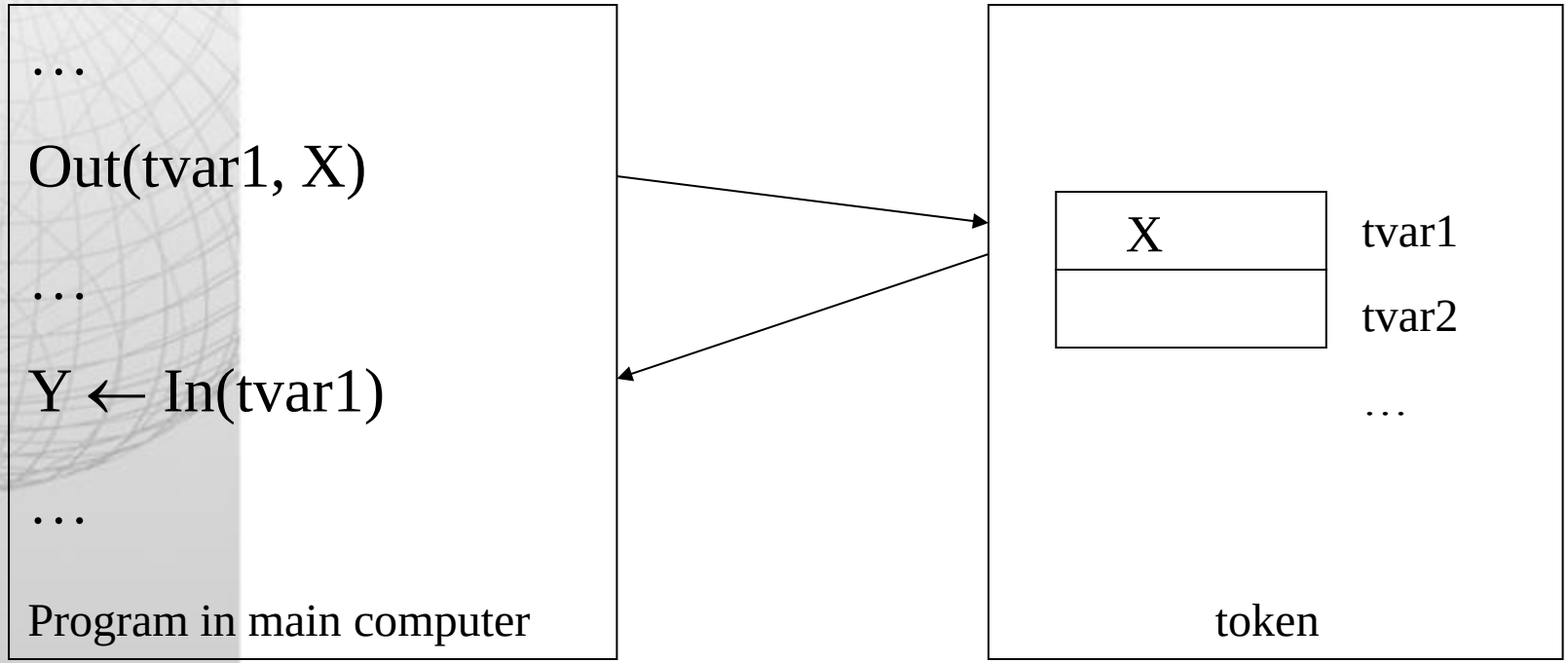
2.1 Variable

- Make some variables reside inside the token

  => For a pirate, it is difficult to see and/or modify the variables residing inside a token.

  See document « variable » for more details

2.1 Variable



Accessing data located in a token variable

2.2 Temporal dissociation

- Break any action to the token into several sub-actions and intermix those sub-actions with other unrelated sub-actions.

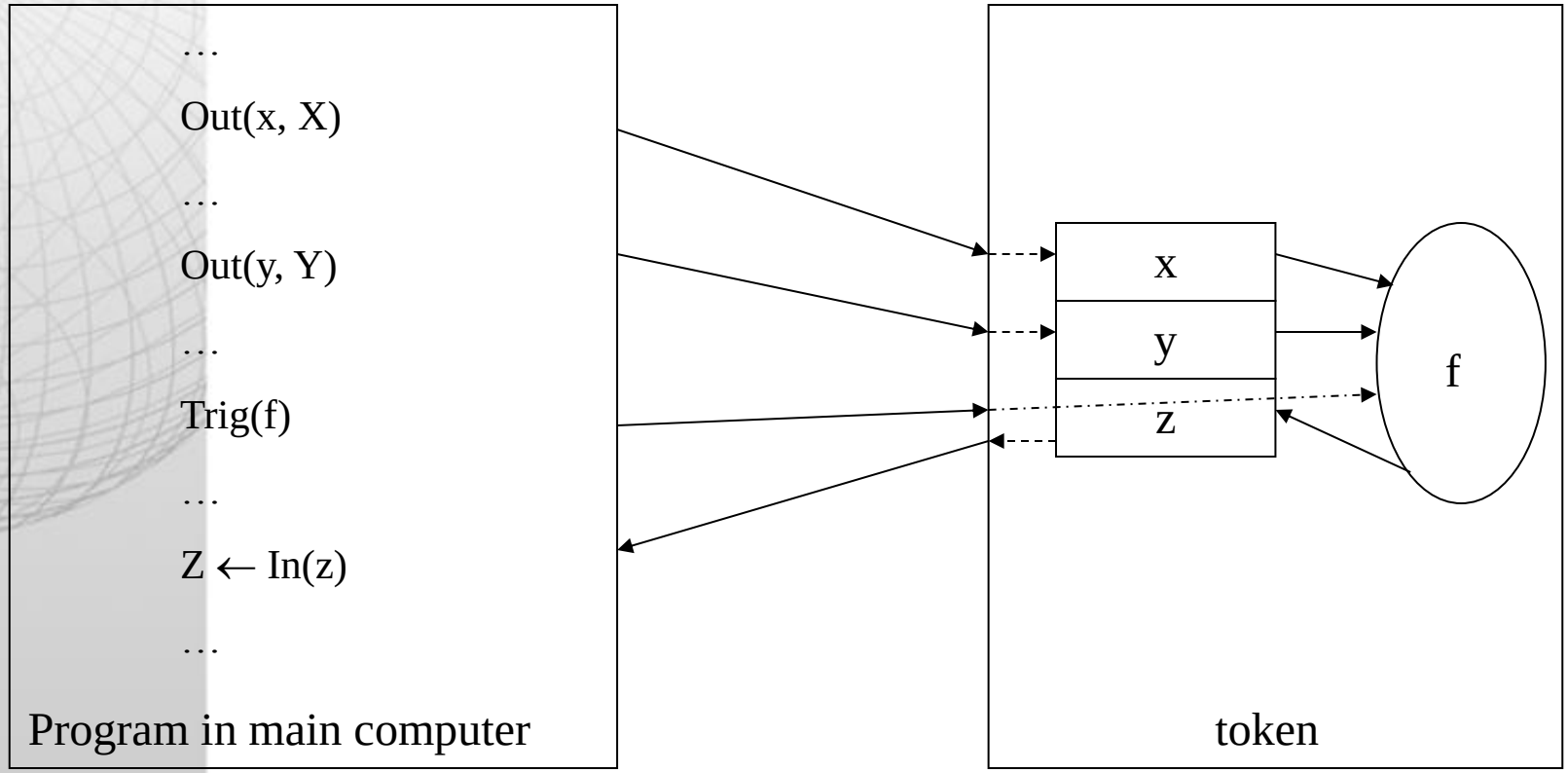
  => For a pirate, the logic of the actions coming in and out of the token is obscured

See document « temporal dissociation » for more details

IT AND ECONOMIC SECURITY

www.validy.com

2.2 Temporal dissociation



Dissociating transfer from execution for function f(x,y)

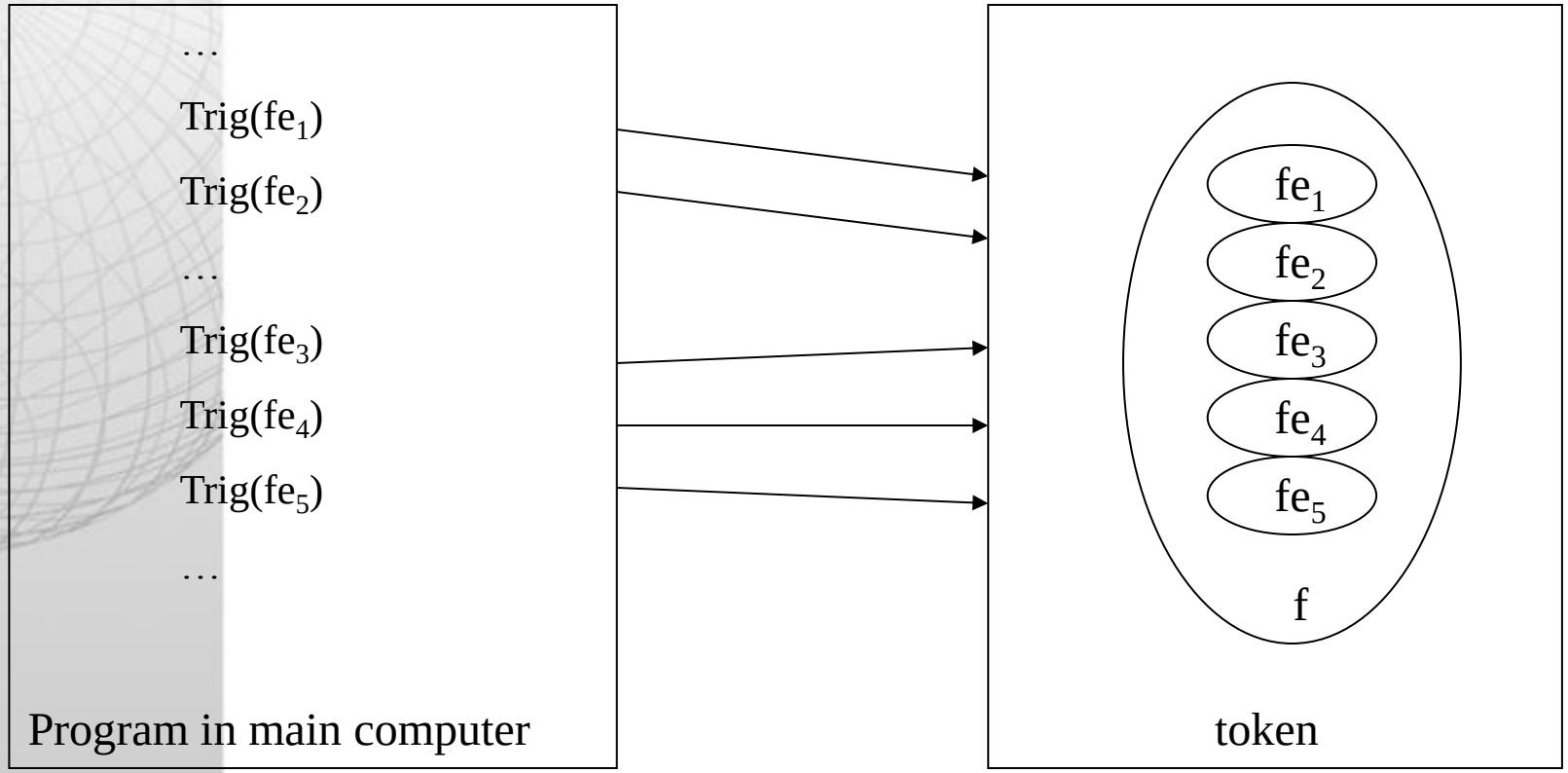
2.3 Elementary functions

- Consider the token as a kind of « coprocessor » and modify its state with several elementary functions (i.e. Add, Move, …)

  => The program can efficiently modify the token state with a few elementary functions.
  They can be pipelined for efficiency.

  See document « elementary functions » for more details

2.3 Elementary functions



Function f is executed with several elementary functions fe_n

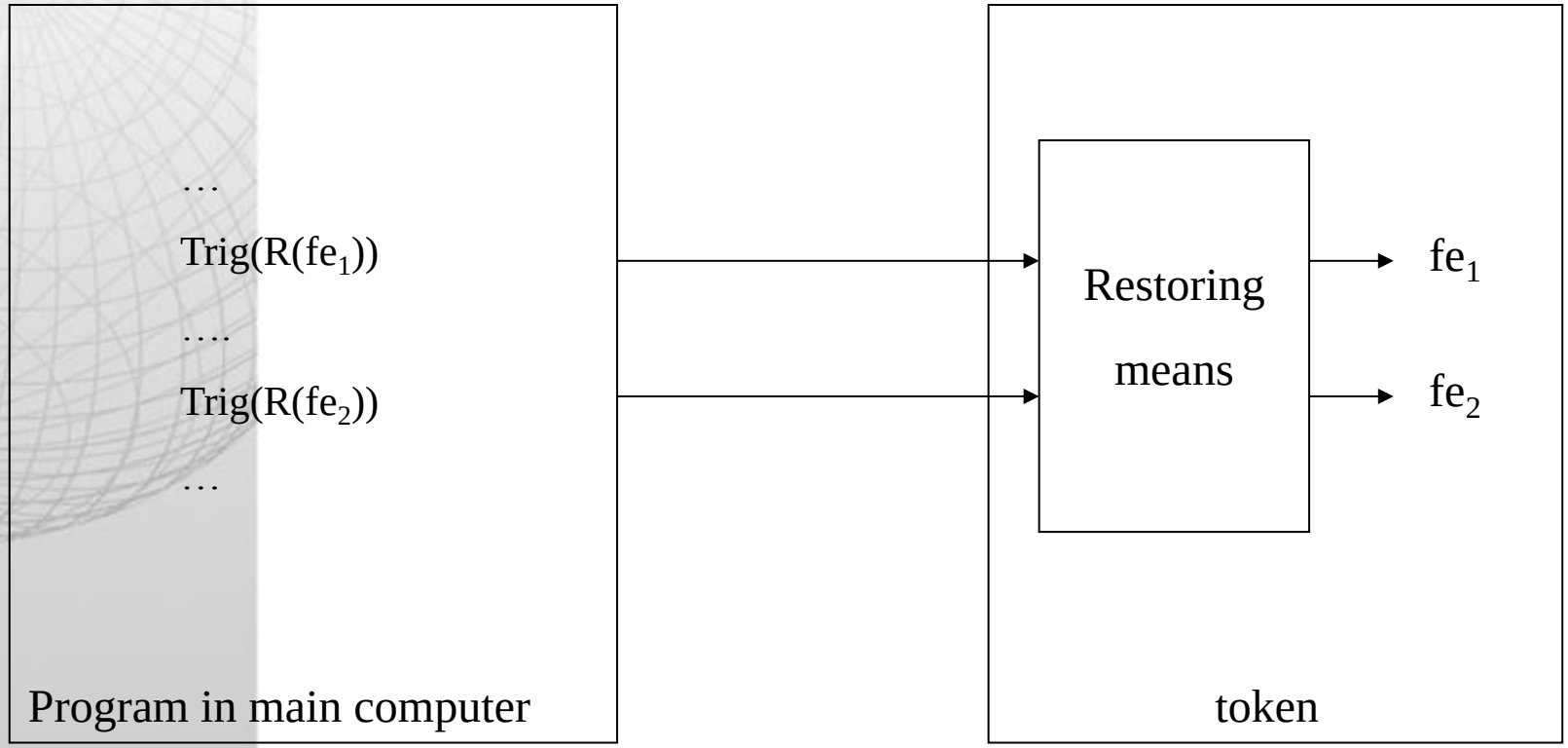
2.4 Renaming

- Rename the commands to the token in order to remove any meaning from them

  =>A pirate cannot tell from a command the type of action performed inside the token.

See document « renaming » for more details

2.4 Renaming



Sending renamed commands to the token

2.5 Detection and coercion

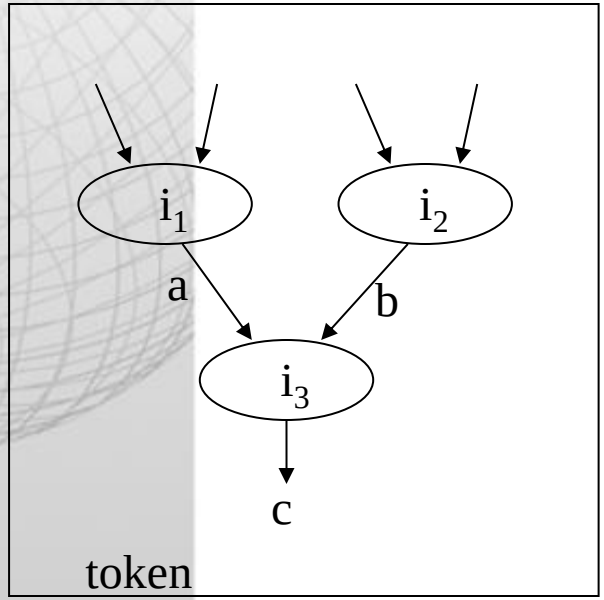
- The token monitors the behavior of the program and **detects** non standard executions by:
  - Verifying the linkage between actions to prevent program modifications and « trial and error » learning approaches.

2.5 Detection and coercion

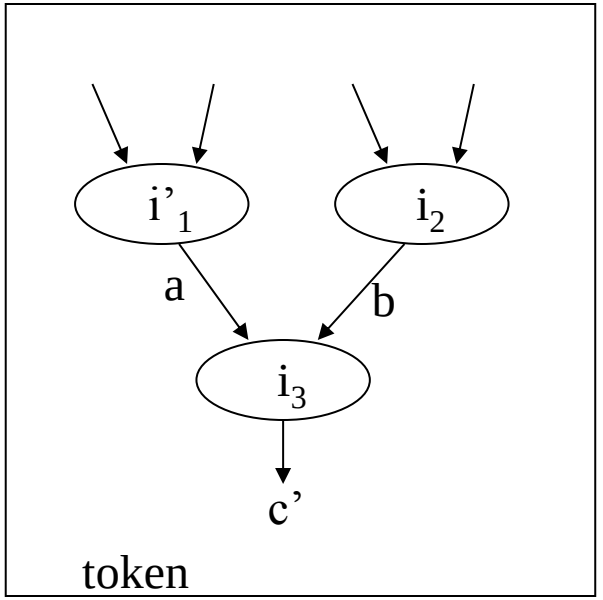
- The token has real **retaliation** power.
- For instance:
  - Mild — Output a warning flag
  - Medium — Stop working for 10 mn
  - High — Self-Destruction

See document « detection and coercion » for more details

2.5 Detection and coercion



Correct instruction chaining



Modified instruction chaining

Execution modification if instruction chaining is modified

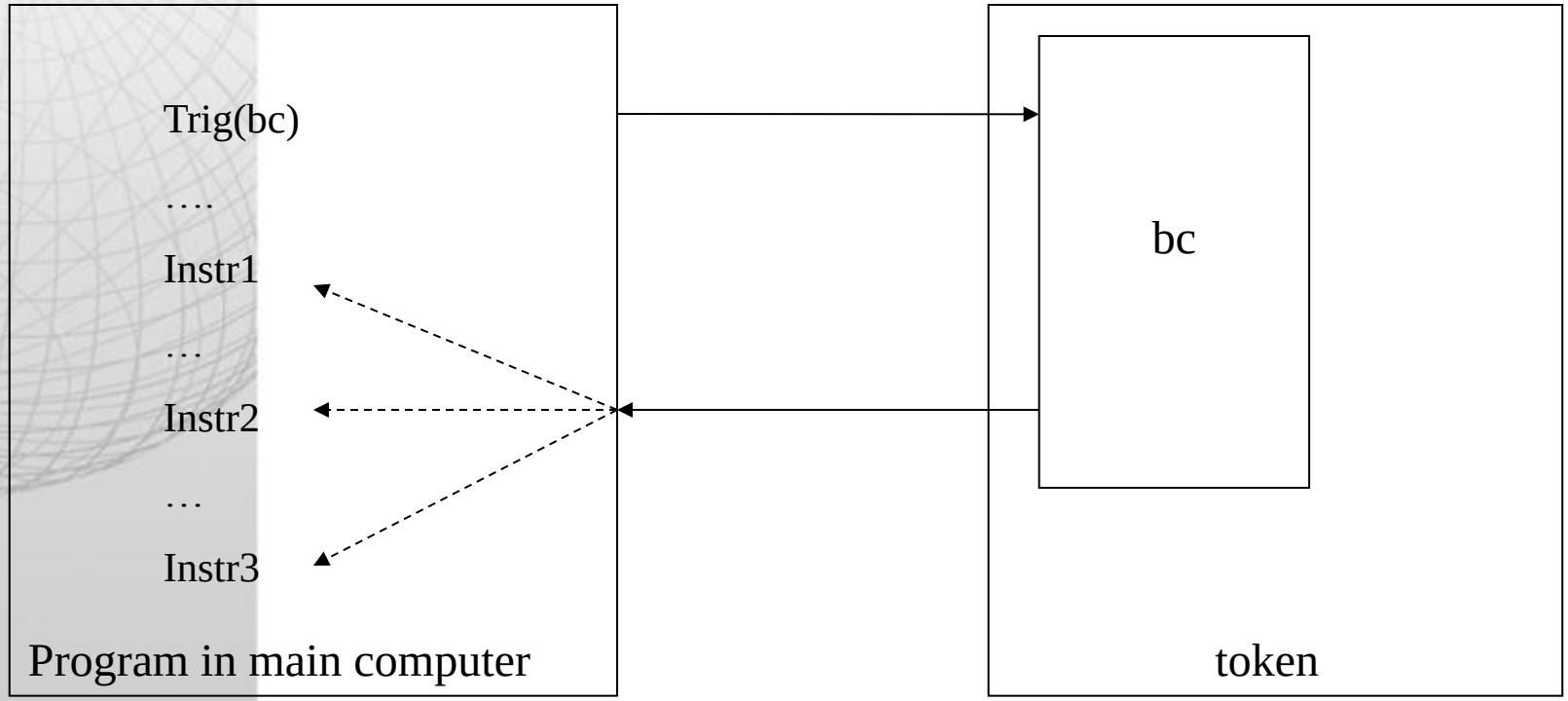
2.6 Conditional branches

- The token can keep state variables used for some conditional branches, perform the computation associated with those branches and output only the results.

  ⇒ A good part of the program structure can be removed from the pirate's view.

See document « conditional branch » for more details

2.6 Conditional branches



The token decides where to continue execution

3 Proof Of Concept

- Based on the 6 foundation principles
  - Programming model
  - Token
  - Protection mechanisms

3.1 Programming model

- The token can implement all the standard resources required by modern programming environments:
  - Registers
  - Stack
  - Heap
- The stack and some of the registers can be thread specific
- Resources can be virtualized as long as data is encrypted by a session key before being paged out.

3.2 Token

- The token is a microcontroller.
- It is programmed to behave as a (security) coprocessor
- It implements registers
- Most instructions use 3 registers : 2 source and 1 destination
- The instruction set uses “Tags” to detect tampering

  Example assembly instructions:

  LDI R1<tag1> 0

  ADD R3<tag3> R1<tag1> R2<tag2>

- Instructions to “join” tags

  LDI R1<tag1> 0

  …

  TGMV R2<tmpTg> R1<tag1> <tag2>

  ADD R1<tag2> R2<tmpTg> R3<tagR3>

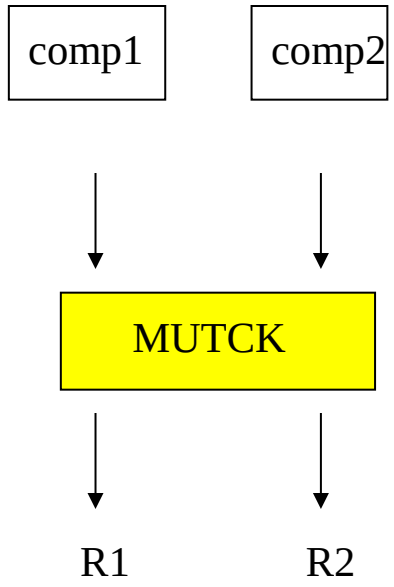
IT AND ECONOMIC SECURITY

www.validy.com

3.3 Mutual protection

- It is possible to link the successful execution of a computation to the successful execution of another

  R1<tag1> ← comp1
  R2<tag2> ← comp2
  XOR R3<tagTmp> R2<tag2> R2<tag2>
  ADD R1<newTag1> R1<tag1> R3<tagTmp>



- This mechanism can also mutually protect two (or more) computations

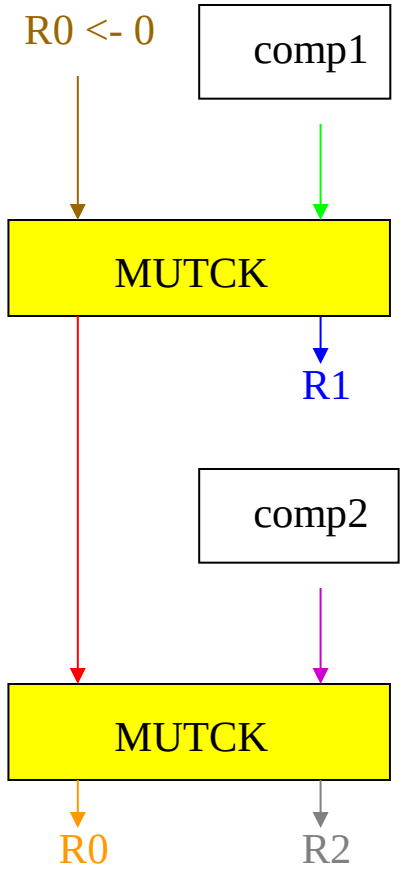
  MUTCK R1<newTag1> <tag1> R2<newTag2> <tag2>

IT AND ECONOMIC SECURITY
www.validy.com

3.4 Link Register and Serial Protection

- One of the “computations” can be to initialize a register with any value
- Then use this register to “link” all the other computations:

```
LDI       R0 <t0> 0
R1 <t1>   <-  comp1
MUTCK     R0 <t10> <t0> R1 <t11> <t1>
R2 <t2>   <-  comp2
MUTCK     R0 <t20> <t10> R2 <t12> <t2>
```



3.5 Locking of additive computations

- Suppose computations comp1, …compN are serially protected.
- If compN is a mandatory computation then comp1,… compN-1 also become MANDATORY

- GREAT for integrity checks

3.6 Calling graph protection

- A function can maintain a link register and serially protect all computations performed in the token (using “join” instructions when needed)
- A calling function can verify execution of a called function by passing a link register to the called function and checking it afterwards
- A called function can verify the identity of its calling function by verifying the tag of its link register

3.7 Cryptography

- The token can implement cryptographic instructions
- Same protection rules:
  - A cryptographic computation, linked to a mandatory “ordinary” computation, becomes mandatory
  - An “ordinary” computation, linked to a mandatory cryptographic computation, becomes mandatory

3.8 Mutual protection of programs

- If two programs using tokens communicate, they can mutually protect each other:

  Using the dialog, each program periodically send a cryptographic challenge to the other program and stop working if the response is inappropriate
  (~ external authenticate)
- A server can protect its clients
- A live-update server can protect remote applications
- ...

IT AND ECONOMIC SECURITY

www.validy.com

3.9 Other Advantages

- The token insures the materiality of the license, it allows a controlled diffusion of the program.
- No need to secure the operating system and/or the hardware.
- No need to have an operating system, even embedded programs can be protected
- Several programs can be protected by the same token.
- Validy Technology security is not based on secrecy. All the principles and the pieces of the implementation can be known and audited.
- ...

IT AND ECONOMIC SECURITY

www.validy.com

4 Program protection

- The majority of Validy Technology benefits can be obtained with an appropriate compiler
  - Automatic generation of token instructions
  - Automatic generation of tags
  - Mutual protection
  - …

IT AND ECONOMIC SECURITY

www.validy.com

4.1 Development process

- We have implemented a Bytecode to Bytecode (re-)compiler to enable the protection of Java programs
- Other compilers are doable
- When a variable is marked to reside in the token, the compiler takes care of all the necessary tasks

4.2 Debug process

- A program is first compiled with the standard compiler and debugged in the standard way
- Then the program is (re-)compiled with the Validy compiler using a test key and tested with a token simulator
- Lastly the program is (re-)compiled with the Validy compiler using a real key and tested with a real token

IT AND ECONOMIC SECURITY

www.validy.com