VIRUS MACHINES :
A new computing paradigm

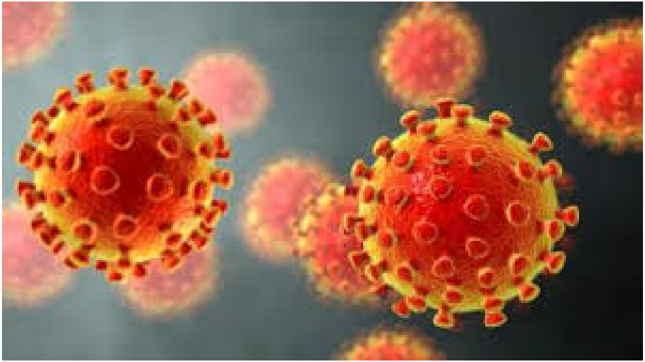
Mario J. Pérez-Jiménez

Research Group on Natural Computing
Dpt. Computer Science and Artificial Intelligence
University of Seville, Spain
Academia Europaea (The Academy of Europe)

www.cs.us.es/~marper marper@us.es

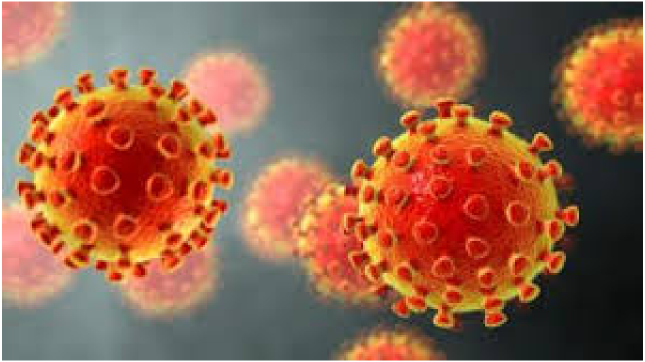
19th Brainstorming Week on Membrane Computing
Sevilla, Spain, January 24-27, 2023

Viruses



Small parasitic biological agents that cannot reproduce by itself.

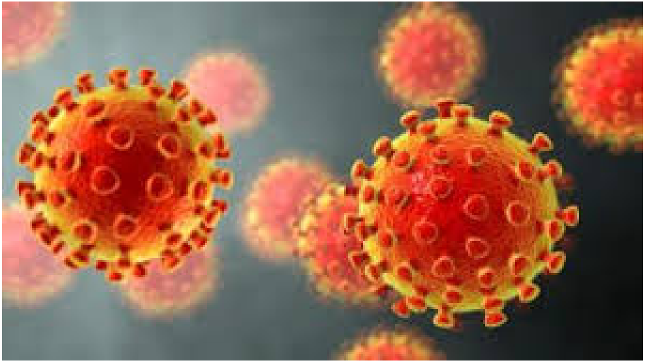
Viruses



Small parasitic biological agents that cannot reproduce by itself.

- ⋆ The most abundant parasites on Earth.

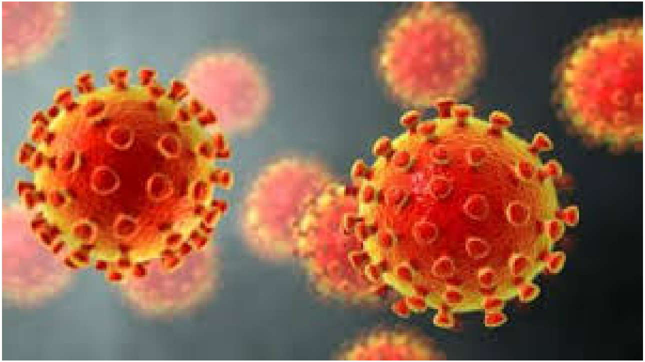
Viruses



Small parasitic biological agents that cannot reproduce by itself.

- ⋆ The most abundant parasites on Earth.
- ⋆ They have not **independent** life (can only inhabit host species).

Viruses

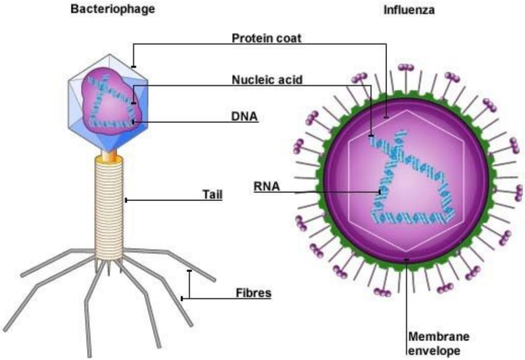


Small parasitic biological agents that cannot reproduce by itself.

- ⋆ The most abundant parasites on Earth.
- ⋆ They have not **independent** life (can only inhabit host species).
- ⋆ Viruses are not lone "wolves". They have **social lives**.

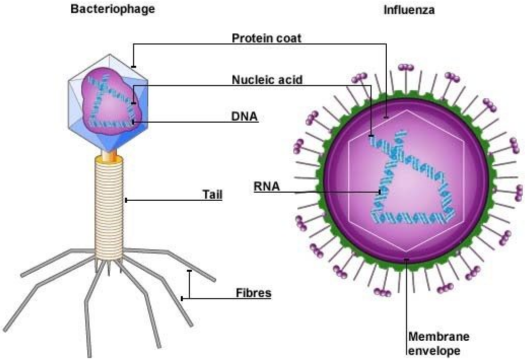
Viruses

A simple structure:



Viruses

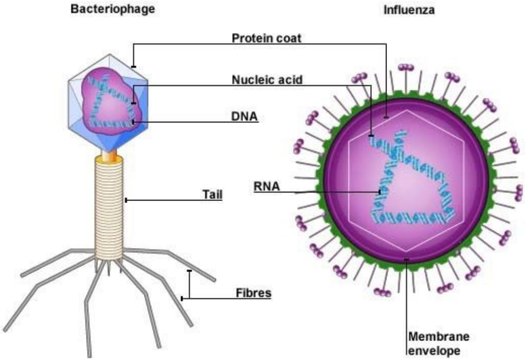
A simple structure:



⋆ Genetic material: either RNA or DNA.

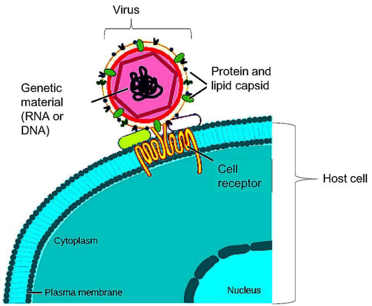
Viruses

A simple structure:

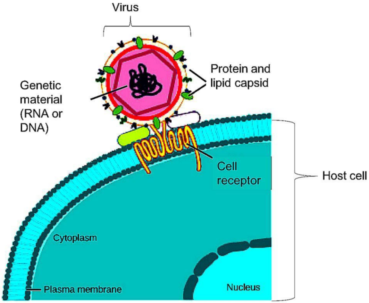


- ⋆ Genetic material: either RNA or DNA.
- ⋆ A protective protein coat.

Viruses

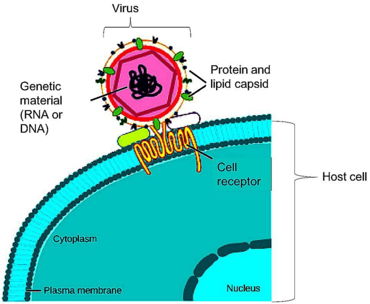

Virus
Genetic material (RNA or DNA)
Protein and lipid capsid
Cell receptor
Host cell
Cytoplasm
Plasma membrane
Nucleus

Viruses



Viruses that infect bacteria (**phage**) have mechanisms that inform them about the possibility of remaining inactive or attacking (depending on new victims).

Viruses



Viruses that infect bacteria (**phage**) have mechanisms that inform them about the possibility of remaining inactive or attacking (depending on new victims).

These processes are active: the phages seem to just sit back and listen in, waiting for bacterial signals to reach some threshold before taking action.

Viral replication

Three phases:

Viral replication

Three phases:

⋆ **Initiation of infection**:

Viral replication

Three phases:

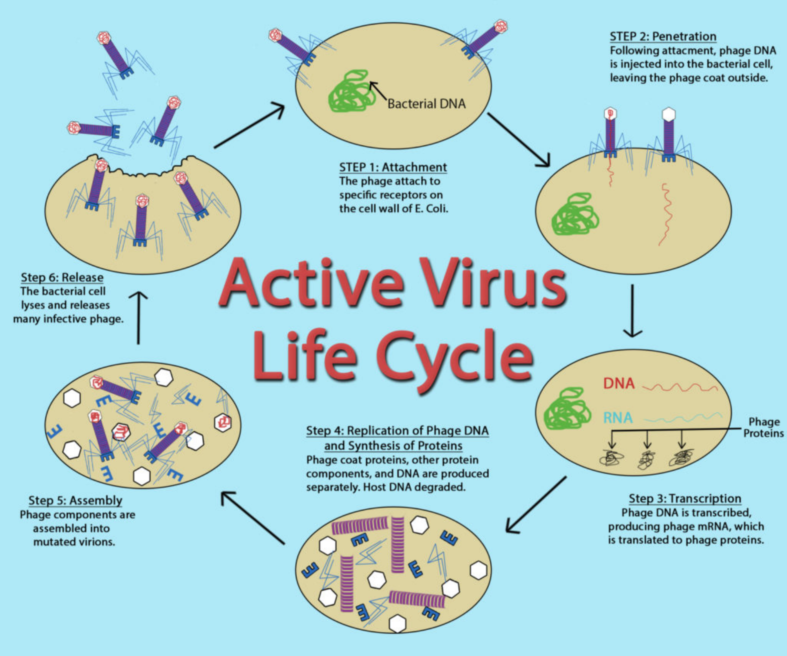
- ⋆ **Initiation of infection**:
- ⋆ **Replication and expression of the genome**.

Viral replication

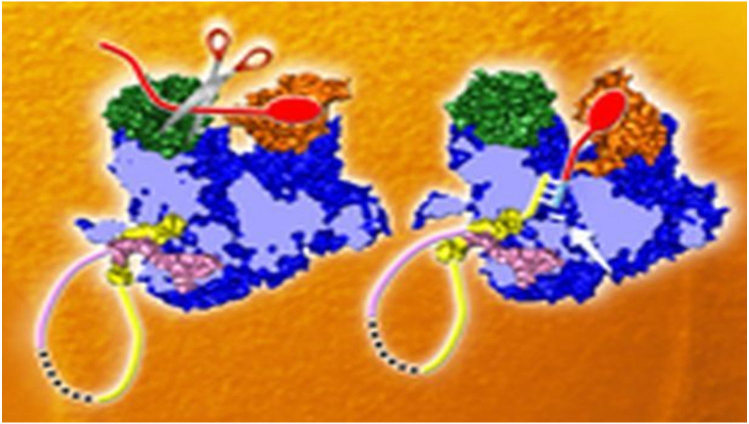
Three phases:

- ⋆ **Initiation of infection**:
- ⋆ **Replication and expression of the genome**.
- ⋆ **The release of the nature virions from the infected cell**.

Viral replication



Virus machines



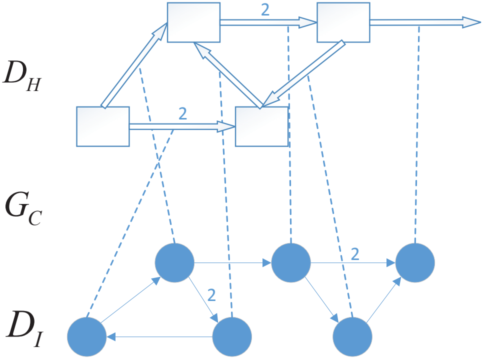
Virus machines

A new computing paradigm inspired by the manner in which viruses transmit from one host to another (introduced in 2015[1]).

[1] L. Valencia, M.J. Pérez-Jiménez, X. Chen, B. Wang, X. Zheng. Basic virus machines. In J.M. Sempere and C. Zandron (eds) **Proceedings of the 16th International Conference on Membrane Computing (CMC16)**, 17-21 August, 2015, Valencia, Spain, pp. 323-342.

Virus machines

A new computing paradigm inspired by the manner in which viruses transmit from one host to another (introduced in 2015[1]).



[1] L. Valencia, M.J. Pérez-Jiménez, X. Chen, B. Wang, X. Zheng. Basic virus machines. In J.M. Sempere and C. Zandron (eds) **Proceedings of the 16th International Conference on Membrane Computing (CMC16)**, 17-21 August, 2015, Valencia, Spain, pp. 323-342.

Virus machines

A VM of degree (p, q), $p \geq 1, q \geq 1$: $(\Gamma, H, I, D_H, D_I, G_C, n_1, \ldots, n_p, i_1)$

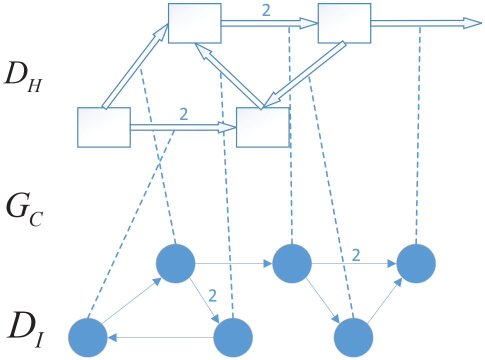
- ⋆ $\Gamma = \{v\}$ is the singleton alphabet (v is called *virus*).
- ⋆ $H = \{h_1, \ldots, h_p\}$, $I = \{i_1, \ldots, i_q\}$ such that $v \notin H \cup I$ and $H \cap I = \emptyset$.
- ⋆ $D_H = (H \cup \{h_0\}, E_H, w_H)$ is a weighted directed graph: $E_H \subseteq H \times (H \cup \{h_0\})$, and w_H is a mapping from E_H onto $\mathbf{N} \setminus \{0\}$.
- ⋆ $D_I = (I, E_I, w_I)$ is a weighted directed graph, where $E_I \subseteq I \times I$, w_I is a mapping from E_I onto $\mathbf{N} \setminus \{0\}$, and for each vertex $i_j \in I$ the out-degree of i_j is ≤ 2.
- ⋆ $G_C = (V_C, E_C)$ is an undirected bipartite graph, where $V_C = I \cup E_H$ being $\{I, E_H\}$ the partition associated with it. In addition, for each vertex $i_j \in I$, the degree of i_j is less than or equal to 1.
- ⋆ $n_j \in \mathbf{N}$ $(1 \leq j \leq p)$ and $i_1 \in I$.

Virus machine of degree (p, q), $p \geq 1, q \geq 1$

A Virus machine of degree (p, q), $p \geq 1, q \geq 1$ can be viewed as:

- A set of p *hosts* labelled with $h_1, \dots, h_p$ and h_j initially contains exactly n_j *viruses*. Symbol h_0 represents the environment of the system.
- A set of q *control instructions* labelled with $i_1, \dots, i_q$.
- Arcs from graph D_H represent *transmission channels* through which viruses can transmit from one host to another: if $(h_r, h_s) \in E_H$ and $w_{r,s}$ is its weight, then $w_{r,s}$ viruses may transmit from host h_r to host h_s (the virus may replicate itself while transmitting). If $s = 0$ then viruses may exit to the environment.
- Each channel is *closed* by default until it is opened by a control instruction (attached to the channel by an edge to G_C) when the instruction is activated.
- Arcs from graph D_I represent *instruction transfer paths* wherein each arc $(i_t, i_{t'}) \in E_I$ is assigned with a weight denoted by $w_{t,t'}$.
- G_C represent the *instruction-channel network* by which an edge $\{i_j, (h_r, h_s)\}$ indicates a control relationship between instruction i and channel (h_r, h_s).

A Virus Machine of degree $(4, 6)$



Virus machines

The virus machines are equivalent in power to Turing machines[2].

[2] X. Chen, M.J. Pérez-Jiménez, L. Valencia, B. Wang, X. Zeng. Computing with viruses. **Theoretical Computer Science**, **623** (2016), 146-159.

[3] A. Romero, L. Valencia, M.J. Pérez-Jiménez. Generating Diophantine Sets by Virus Machines. In M. Gong, L. Pan, T. Song, K. Tang, X. Zhang (eds) **Bio-Inspired Computing: Theories and Applications. The 10th International Conference (BIC-TA 2015)**, Hefei, China, September 25-28, 2015. Proceedings, pp. 331-341.

[4] A. Romero, L. Valencia, A. Riscos, M.J. Pérez-Jiménez. Computing partial recursive functions by Virus Machines. **Lecture Notes in Computer Science**, **9504** (2015), 353-368.

Virus machines

The virus machines are equivalent in power to Turing machines[2].

They have the ability to:

- ⋆ **Generate** all **diophantine sets**[3]

[2] X. Chen, M.J. Pérez-Jiménez, L. Valencia, B. Wang, X. Zeng. Computing with viruses. **Theoretical Computer Science**, **623** (2016), 146-159.

[3] A. Romero, L. Valencia, M.J. Pérez-Jiménez. Generating Diophantine Sets by Virus Machines. In M. Gong, L. Pan, T. Song, K. Tang, X. Zhang (eds) **Bio-Inspired Computing: Theories and Applications. The 10th International Conference (BIC-TA 2015)**, Hefei, China, September 25-28, 2015. Proceedings, pp. 331-341.

[4] A. Romero, L. Valencia, A. Riscos, M.J. Pérez-Jiménez. Computing partial recursive functions by Virus Machines. **Lecture Notes in Computer Science**, **9504** (2015), 353-368.

Virus machines

The virus machines are equivalent in power to Turing machines[2].

They have the ability to:

- ⋆ **Generate** all **diophantine sets**[3]
- ⋆ **Compute** all **recursive functions**[4].

[2] X. Chen, M.J. Pérez-Jiménez, L. Valencia, B. Wang, X. Zeng. Computing with viruses. **Theoretical Computer Science**, **623** (2016), 146-159.

[3] A. Romero, L. Valencia, M.J. Pérez-Jiménez. Generating Diophantine Sets by Virus Machines. In M. Gong, L. Pan, T. Song, K. Tang, X. Zhang (eds) **Bio-Inspired Computing: Theories and Applications. The 10th International Conference (BIC-TA 2015)**, Hefei, China, September 25-28, 2015. Proceedings, pp. 331-341.

[4] A. Romero, L. Valencia, A. Riscos, M.J. Pérez-Jiménez. Computing partial recursive functions by Virus Machines. **Lecture Notes in Computer Science**, **9504** (2015), 353-368.

The RSA cryptosystem

[5] R.L. Rivest, A. Shamir, L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. CAMC, **21**, 2 (1978), 120-126.

The RSA cryptosystem

The first public-key cryptosystem to verify the conditions formulated by W. Diffie and M. Hellman

[5] R.L. Rivest, A. Shamir, L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. CAMC, **21**, 2 (1978), 120-126.

The RSA cryptosystem

The first public-key cryptosystem to verify the conditions formulated by W. Diffie and M. Hellman

It was introduced by R. **R**ivest, A. **S**hamir and L. **A**dleman in 1978 [5].

[5] R.L. Rivest, A. Shamir, L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. CAMC, **21**, 2 (1978), 120-126.

The RSA cryptosystem

The first public-key cryptosystem to verify the conditions formulated by W. Diffie and M. Hellman

It was introduced by R. **R**ivest, A. **S**hamir and L. **A**dleman in 1978 [5].

The underlying problem of the **RSA** system is the **semiprime factorization problem**:

[5] R.L. Rivest, A. Shamir, L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. **CAMC**, **21**, 2 (1978), 120-126.

The RSA cryptosystem

The first public-key cryptosystem to verify the conditions formulated by W. Diffie and M. Hellman

It was introduced by R. **R**ivest, A. **S**hamir and L. **A**dleman in 1978 [5].

The underlying problem of the **RSA** system is the **semiprime factorization problem**: "*given a semiprime number, find its decomposition*"

[5] R.L. Rivest, A. Shamir, L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. **CAMC**, **21**, 2 (1978), 120-126.

The RSA cryptosystem

The first public-key cryptosystem to verify the conditions formulated by W. Diffie and M. Hellman

It was introduced by R. **R**ivest, A. **S**hamir and L. **A**dleman in 1978 [5].

The underlying problem of the **RSA** system is the **semiprime factorization problem**: "*given a semiprime number, find its decomposition*"

(Semiprime: the product of exactly two prime numbers).

[5] R.L. Rivest, A. Shamir, L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. CAMC, **21**, 2 (1978), 120-126.

The RSA cryptosystem

The first public-key cryptosystem to verify the conditions formulated by W. Diffie and M. Hellman

It was introduced by R. **R**ivest, A. **S**hamir and L. **A**dleman in 1978 [5].

The underlying problem of the **RSA** system is the **semiprime factorization problem**: "*given a semiprime number, find its decomposition*"

(Semiprime: the product of exactly two prime numbers).

This problem can be characterized by the following partial function FACT: for each semiprime $x = y \cdot z$, with $y \geq z \geq 2$, we have $\mathtt{FACT}(x) = (y, z)$.

[5] R.L. Rivest, A. Shamir, L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. **CAMC**, **21**, 2 (1978), 120-126.

The RSA cryptosystem

The first public-key cryptosystem to verify the conditions formulated by W. Diffie and M. Hellman

It was introduced by R. **R**ivest, A. **S**hamir and L. **A**dleman in 1978 [5].

The underlying problem of the **RSA** system is the **semiprime factorization problem**: "*given a semiprime number, find its decomposition*"

(Semiprime: the product of exactly two prime numbers).

This problem can be characterized by the following partial function FACT: for each semiprime $x = y \cdot z$, with $y \geq z \geq 2$, we have $\mathtt{FACT}(x) = (y, z)$.

The *semiprime factorization problem* is conjectured to be a **computationally hard** problem.

[5] R.L. Rivest, A. Shamir, L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. CAMC, **21**, 2 (1978), 120-126.

The RSA cryptosystem

The first public-key cryptosystem to verify the conditions formulated by W. Diffie and M. Hellman

It was introduced by R. **R**ivest, A. **S**hamir and L. **A**dleman in 1978 [5].

The underlying problem of the **RSA** system is the **semiprime factorization problem**: "*given a semiprime number, find its decomposition*"

(Semiprime: the product of exactly two prime numbers).

This problem can be characterized by the following partial function FACT: for each semiprime $x = y \cdot z$, with $y \geq z \geq 2$, we have $\texttt{FACT}(x) = (y, z)$.

The *semiprime factorization problem* is conjectured to be a **computationally hard** problem.

Any "large" semiprime input n for **RSA** is used as the *modulus* for both public and private keys.

[5] R.L. Rivest, A. Shamir, L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. CAMC, **21**, 2 (1978), 120-126.

The RSA cryptosystem

The first public-key cryptosystem to verify the conditions formulated by W. Diffie and M. Hellman

It was introduced by R. **R**ivest, A. **S**hamir and L. **A**dleman in 1978 [5].

The underlying problem of the **RSA** system is the **semiprime factorization problem**: "*given a semiprime number, find its decomposition*"

(Semiprime: the product of exactly two prime numbers).

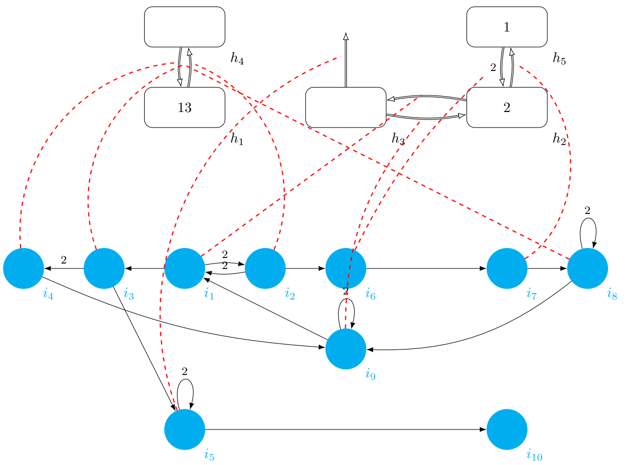
This problem can be characterized by the following partial function FACT: for each semiprime $x = y \cdot z$, with $y \geq z \geq 2$, we have $\mathtt{FACT}(x) = (y, z)$.

The *semiprime factorization problem* is conjectured to be a **computationally hard** problem.

Any "large" semiprime input n for **RSA** is used as the *modulus* for both public and private keys. In order to attack the **RSA** system, the factorization of n is needed.

[5] R.L. Rivest, A. Shamir, L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. CAMC, **21**, 2 (1978), 120-126.

A VM computing the partial function FACT



i_1 activates $h_2 \to h_3$

i_2 activates $h_1 \to h_4$

i_3 activates $h_1 \to h_4$

i_4 activates $h_4 \to h_1$

i_5 activates $h_3 \to env$

i_6 activates $h_5 \to h_2$ (2)

i_7 activates $h_2 \to h_5$

i_8 activates $h_4 \to h_1$

i_9 activates $h_3 \to h_2$

THANK YOU

FOR YOUR ATTENTION!