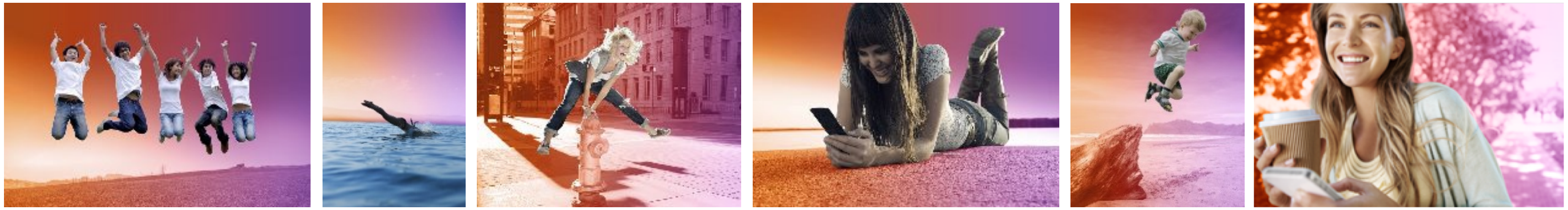


VIACCESS-ORCA CRYPTOCOMP
Homomorphic Ad Server

B.MOUCHIR AWAD/ D.RENARD

31-may-2018

Agenda

1. Introduction:

<u>What is CRYPTOCOMP?</u>

CRYPTOCOMP is a collaborative project financially supported by the French government and composed by several French cryptographic laboratories (INRIA,CEA-LIST, UVSQ, ENS, XLIM, CNRS, Lab-STICC) and industrial partners (Idemia, CryptoExperts, Kalray, Bertin Technologie, Gemalto, Viaccess-Orca) which aims at providing demonstrators in Full Homomorphic Encryption.

<u>VIACCESS-ORCA use case:</u>

The Viaccess-Orca version of CRYPTOCOMP is **a Targeted Advertising System** which use Full Homomorphic Encryption (F.H.E.) libraries from CEA-LIST for **ciphering user data** and proposing a privacy data protection.

<u>2018 New European General Data Protection Regulation :</u>

Since May 25th of 2018, new European General Data Protection Regulation (**GDPR / Regulation (EU) 2016/679**) are applicable which introduces many changes in the definition of personal data , their mandatory protection, and the amount of possible financial penalty up to 4 % of the total worldwide annual revenue.

2. General Description of the System

<u>2.1. Presentation</u>

The Viaccess-Orca CRYPTOCOMP application is made of three parts:

- a Host Server hosting an Online Magazine website
- an Advertising Server
- a Client Web Browser

The technology chosen is **Node-JS** for Server and **JavaScript and CHROME ADD-ON** for Browser.

<u>Functional principle of the protection of user data</u>

Each time a new user goes to the VO CRYPTOCOMP magazine web site, he registers and gets a user and password.

Then, once logged into the system, he is directed to the Homepage where he has four possible choices of articles to read depending of his interest (Travel, Food, Sport, Music).

As soon as the user clicks on a articles category, he gets both a link of articles to read and an advertising targeted to the category chosen.

2.2. The different phases of the project

Viaccess-Orca have been working on the project for 3 years approximately with around five times 6 month design periods with Partnership programs:

- **Cryptocomp v1 : Homomorphic Health IOT Server**
  => the development of Health IOT & Android Client/Server let us improve our understanding of a Homomorphic possible use case.

- **Cryptocomp v2 : Homomorphic Ad Server (Feasibility phase)**
  => Feasibility studies of Interfacing Technics with CEA libraries, Advertising market, GDPR regulations, and possible diagrams of Homomorphic Ad Server Architecture.

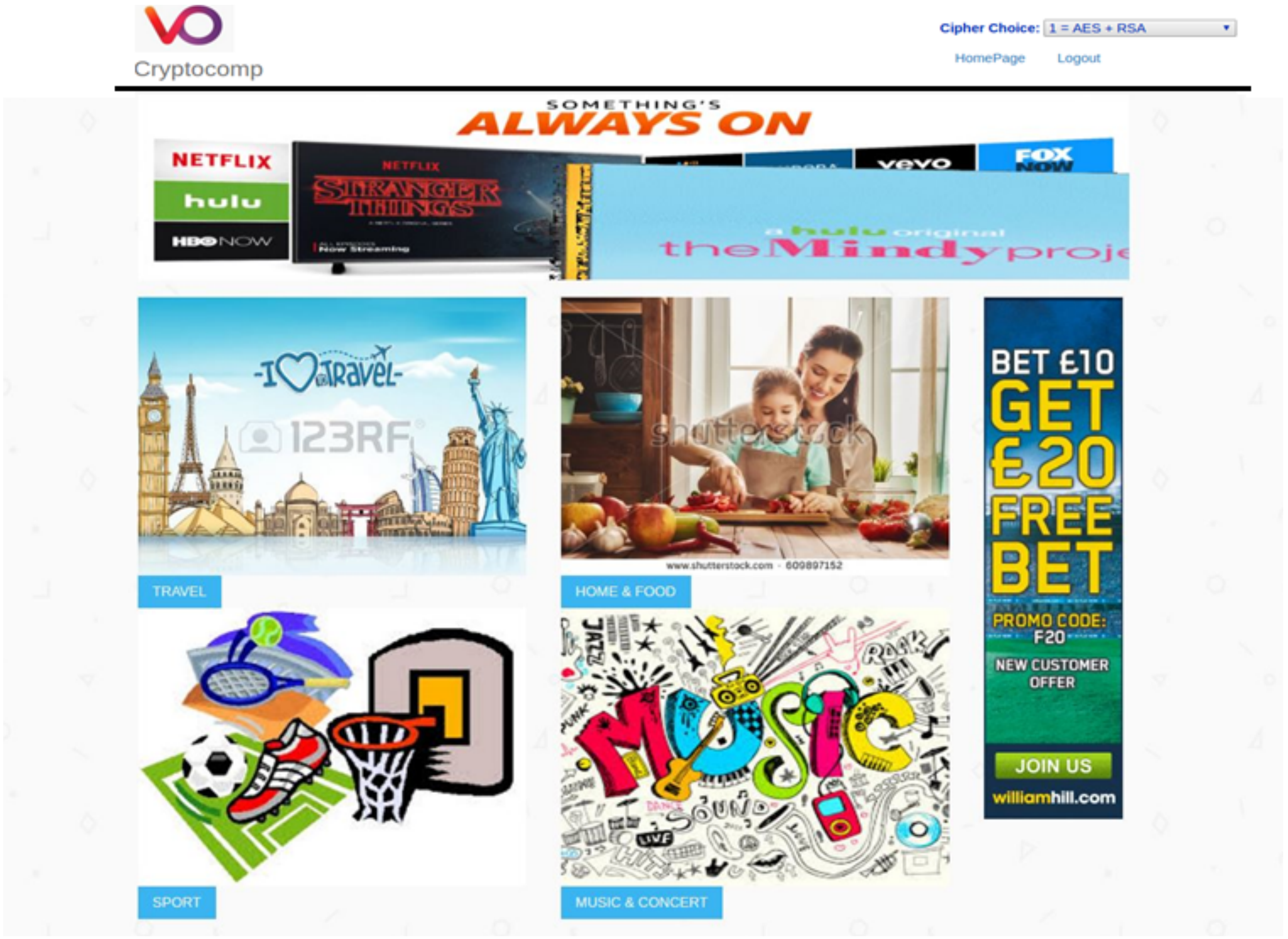
- **Cryptocomp v3 : Homomorphic Ad Server (Initial Development phase)**
  => Specifications and flow chart of Node-JS /JavaScript Client /Servers
  => Development of standalone version of **Simple Homomorphic Ad server with RSA** instead of **FHE**, in order to prepare CEA libraries integration.

- **Cryptocomp v4 : Homomorphic Ad Server (Improvement phase)**

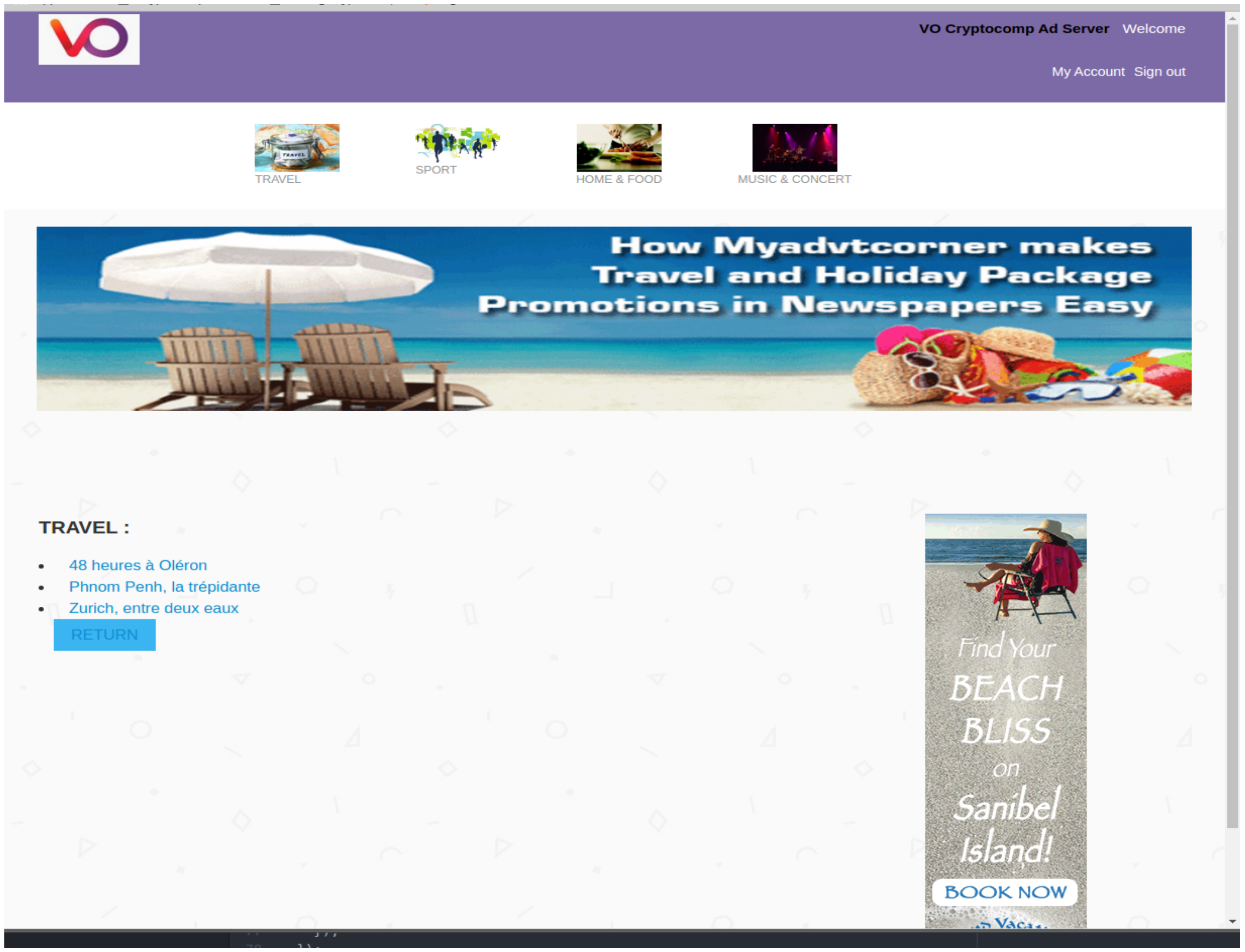
- **Cryptocomp v5 : Homomorphic Ad Server (integration phase of CEA libraries) (End scheduled in Sept2018)**
  => Refactoring JavaScript http Request/answer to integrate call to binaries
  => creation of binaries with Armadilo/Cingulata Compiler Toolchain.

.

3.1. Home page of VIACCESS-ORCA CRYPTOCOMP Web magazine



3.2. Articles List Page of Web magazine



VO Cryptocomp Ad Server Welcome

My Account Sign out

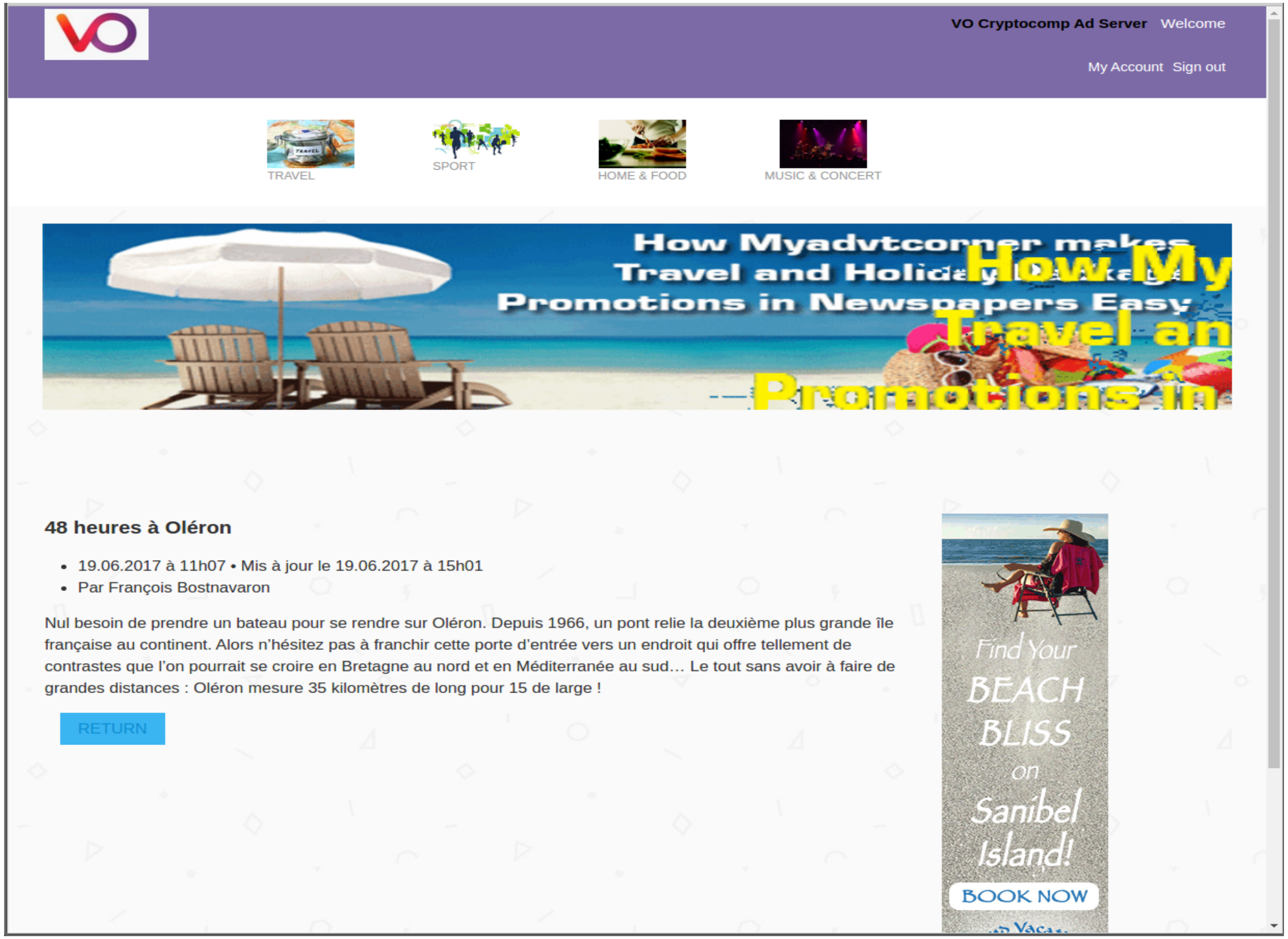
TRAVEL
SPORT
HOME & FOOD
MUSIC & CONCERT
How Myadvtcorner makes Travel and Holiday Package Promotions in Newspapers Easy
Find Your BEACH BLISS on Sanibel Island!
BOOK NOW

TRAVEL :

- 48 heures à Oléron
- Phnom Penh, la trépidante
- Zurich, entre deux eaux

RETURN

3.3. Travel Article Page of Web magazine



VO Cryptocomp Ad Server Welcome

My Account Sign out

TRAVEL SPORT HOME & FOOD MUSIC & CONCERT

How Myadvtcorner makes Travel and Holiday Package Promotions in Newspapers Easy
How My Travel an Promotions in
Find Your BEACH BLISS on Sanibel Island!
BOOK NOW

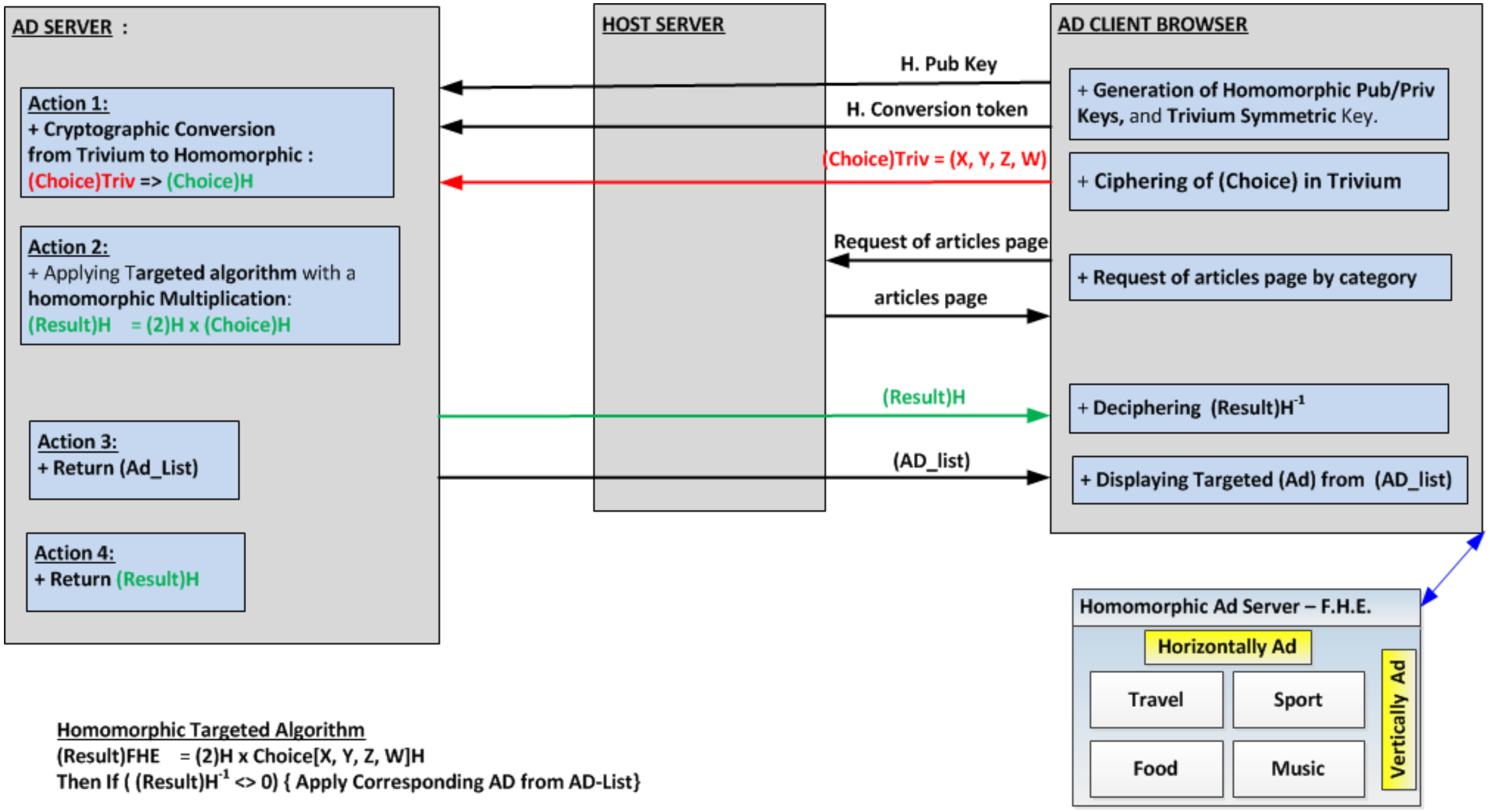
48 heures à Oléron

- 19.06.2017 à 11h07 • Mis à jour le 19.06.2017 à 15h01
- Par François Bostnavaron

Nul besoin de prendre un bateau pour se rendre sur Oléron. Depuis 1966, un pont relie la deuxième plus grande île française au continent. Alors n'hésitez pas à franchir cette porte d'entrée vers un endroit qui offre tellement de contrastes que l'on pourrait se croire en Bretagne au nord et en Méditerranée au sud… Le tout sans avoir à faire de grandes distances : Oléron mesure 35 kilomètres de long pour 15 de large !

RETURN

4. Architecture diagram of Full Homomorphic Encryption Ad Server


AD SERVER :
Action 1:
+ Cryptographic Conversion from Trivium to Homomorphic :
(Choice)Triv => (Choice)H
Action 2:
+ Applying Targeted algorithm with a homomorphic Multiplication:
(Result)H = (2)H x (Choice)H
Action 3:
+ Return (Ad_List)
Action 4:
+ Return (Result)H
HOST SERVER
AD CLIENT BROWSER
H. Pub Key
H. Conversion token
(Choice)Triv = (X, Y, Z, W)
Request of articles page
articles page
(Result)H
(AD_list)
+ Generation of Homomorphic Pub/Priv Keys, and Trivium Symmetric Key.
+ Ciphering of (Choice) in Trivium
+ Request of articles page by category
+ Deciphering (Result)H-1
+ Displaying Targeted (Ad) from (AD_list)
Homomorphic Ad Server – F.H.E.
Horizontally Ad
Vertically Ad
Travel
Sport
Food
Music

Homomorphic Targeted Algorithm
(Result)FHE = (2)H x Choice[X, Y, Z, W]H
Then If ((Result)H^{-1} <> 0) { Apply Corresponding AD from AD-List}

5. Description of Ciphering Steps of the user personal Data

- Step 1: User Registration:
  => generation of login/password
  => generation of symmetric key (Trivium)
  => generation of Asymmetric Public/Private Keys (FHE)
  => generation of the Homomorphic Conversion token

- Step 2: Initialization of the user choice variable by choosing one category of articles to read of the four:
  => Choice[X,Y,Z,W] becomes Choice[5,0,0,0] if the user choose "Travel".

- Step 3: Ciphering the user choice variable into symmetric (Trivium), and Sending to Ad Server.
  => (Choice)*

- Step 4 : Reception of the (Choice)* ciphered in (Trivium) by the Ad server and convert it into an homomorphic data: => (Choice)H

- Step 5 : Applying the targeted algorithm on to the homomorphic (Choice)H, and sending back to the user: => (Result)H = (Constante)H x Choice[X, Y, Z, W]H

- Step 6 : Reception of both a list of four advertising and an homomorphic (Result)H, and display only one of the four , thanks to the content of the Result:
  => Then If ((Result)H-1 <> 0) { Apply Corresponding AD from AD-List}

6. Presentation of the Homomorphic Cryptography of the Project

6.1. Introduction:

Since the breakthrough of **Craig Gentry in 2009** about **fully homomorphic Encryption**, with the possibility of being able to process calculation over encrypted data, many studies have been published for improving the performance and the limitation of the schemes.

One of the scheme called "**Bootstrapping**", introduced by Gentry, enables to reduce the noise after an homomorphic calculation.

And other one is called "**Somewhat Homomorphic Encryption**" enables a homomorphic calculation with a limited noise by choosing a crypto system with a multiplicative depth small and known.

This project use the implementation of SWHE scheme developed by the **CEA-LIST in the Cingulata Compiler Toolchain.**

6.2. Homomorphic Cipher-text compression mechanism

Instead of **encrypting a data (m) in Homomorphic**, the system split the operation in **two parts**:

- Firstly the data is **encrypted in symmetric (Trivium)** => **Ek(m)**
- And then a Cryptographic conversion Token is generated (which is in fact a Trivium key encrypted in Homomorphic). => **HEpk(k)**

Then thanks to the **decompression circuit** of the **CINGULATA compiler Toolchain library,** the final data is decompressed and transformed in **homomorphic**. => **c = HEpk(m)**

By this method, the variable part of the data is small and is easy to transport, rather than the other is big but is calculated only one time, and need to be sent only at the beginning of the communication.

6.3. Advantages of the technic

- **The user data remains secret** compare to Ad Server.
- **The size of the ciphered Data transported is reduced** with the Trivium, and still have it at the end in Homomorphic format (FHE) for being used in calculations.

6.4. Difference between Simple and Full Homomorphic Encryption:

A Simple Homomorphic Encryption means only one Homomorphic operation at a time is supported, Addition or multiplication, rather than a Full Homomorphic Encryption (FHE) allow both operations at the same time.

RSA is simple Homomorphic with the Multiplication.

Example of Homomorphic addition:

- E = Enc(a)$\boxed{+}$Enc(b)   Dec(E) = a+b
  $\boxed{+}$ means « addition » in this formula

Example of Homomorphic multiplication:

- E = Enc(a) $\boxed{x}$ Enc(b)   Dec(E) = a x b
  $\boxed{x}$ means « multiplication » in this formula

7. Theoretical framework of Full Homomorphic compression used in project:

Extract from REF-3 :(Stream Cipher: a Practical solution for Effcient Homomorphic Ciphertext Compression)

Suppose that **Alice** wants to send an Homomorphic Encrypted **HE** message **m** to **Charlie**:

c = HEpk(m)

By adding a Trivium symmetric Encryption Scheme **E**, the ciphertext sent to Charlie becomes:

c' = (HEpk(k); Ek(m))

Then Charlie recover the Homomorphic message by Ciphertext **decompression circuit**:

$$c = \mathsf{HE}_{\mathsf{pk}}(m) = \mathcal{C}_{\mathsf{E}^{-1}}\left(\mathsf{HE}_{\mathsf{pk}}(k), \mathsf{E}_k(m)\right)$$

For being able to describe better the decompression**, Ek(m) is**

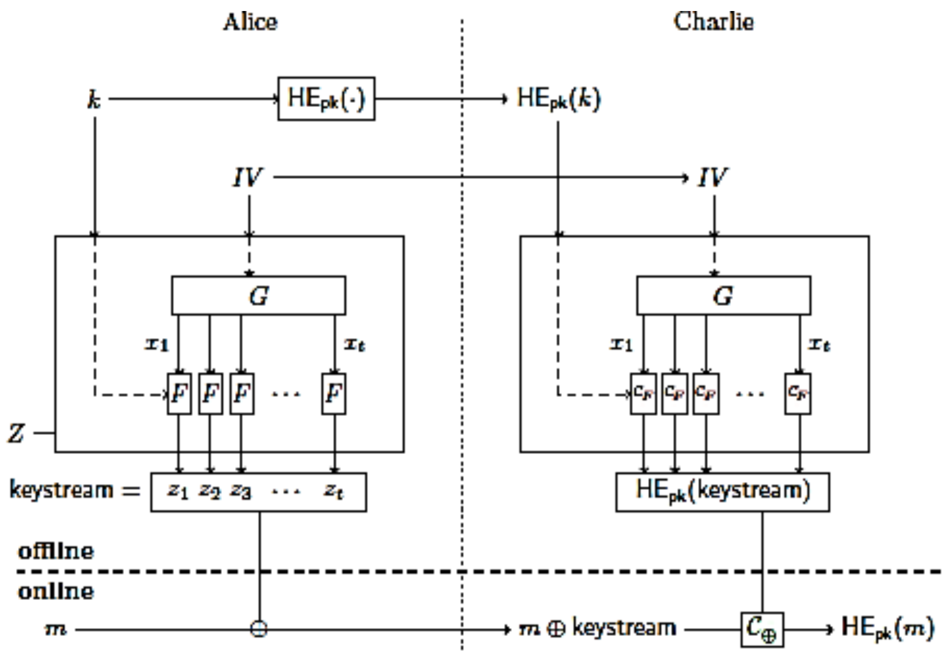
$$\mathsf{E}_k(m) \overset{\text{def}}{=} (IV, \mathsf{E}'_{k,IV}(m))$$

And by adding an **additive IV-based stream cipher Z**, we get:

$$\mathsf{E}'_{k,IV}(m) = Z(k, IV) \oplus m$$

Then the Offline/ Online Ciphertext Decompression steps are:

1. an offline ciphering and sending **HEpk(k)** only once, and **IV**
2. an offline initialization of **keystream** and **HEpk(keystream)**
3. an online decompression by aggregating (**m + keystream**) with **HEpk(keystream)**



8. List of corresponding Homomorphic or Trivium interfaces binary executable

On our architecture, here are the **list of** binaries run from Client or Server for the Homomorphic Cryptography or Trivium:

Web Browser Side: (ciphering)

- Generate_Keys_FHE() => generate FHE Pub & Priv user Keys
- Generate_Key_Trivium() => generate Symmetric Trivium user Key
- Generate_Token() => **cipher the Trivium key with FHE Pub key**
- Init_Keystream() => offline initialize Trivium keystream of the user
- Cipher_Choice_Trivium() => stream cipher the **(Choice)** in Trivium => **(choice + keystream)**

Ad Server Side:

- Init_FHE_Keystream() => offline compute **[HEpk(keystream)]** from the **Token** and **IV**
- Compute_Choice_FHE() => online aggregate **[HEpk(keystream)]** and **(choice + keystream)** for computing **(Choice)FHE**
- Compute_Result_FHE() => do a Multiplication: **R = (Constante)FHE x (Choice)FHE**

Web Browser Side: (deciphering)

- Decipher_Result_FHE() => decipher **(Result_FHE)** and display the targeted advertising.

9. Interfacing with the CEA-LIST Armadillo/CINGULATA Platform

In order to build the homomorphic binary executables, as described earlier, a special Toolchain is used to do that: the **Armadillo/Cingulata compiler Toolchain** developed by the CEA-LIST .

See https://github.com/CEA-LIST/Cingulata

We will use it mainly to:

- Generate **Homomorphic keys**
- Generate the **Homomorphic token**
- Initialize the **HEpk(keystream)**
- And **compute the Homomorphic multiplication** between (Choice)H and a (Cte)H

A separated presentation will introduce all the steps of the Toolchain compilation and clean up calculation up to the final Homomorphic message.

10. Performance and size of Homomorphic Data

10.1. 2016 Experimental Results:

Extract from REF-3 :(Stream Cipher: a Practical solution for Effcient Homomorphic Ciphertext Compression)

Crypto Conversion of a data from Trivium-12 to Homomorphic: with BGV scheme (1 core)

Bits	keystream	x depth	Latency
80	45	12	1417s (23,6 mn)

Crypto Conversion of a data from Trivium-12 to Homomorphic: with FV scheme (48 core server)

Bits	keystream	x depth	Latency 1 core	Latency 48 core
80	57	12	681s (11,3 mn)	27s

Number of gates used by Trivium-12:

FV

#ANDs	length	#XORs	keystream
80	3237	15019	57

BGV

#ANDs	#XORs	keystream
3183	14728	45

10.2. VO CRYPTOCOMP Cingulata Results (=> end of Sept2018)

Trivium-12 algorithm.

- **with key, IV of 80bits** and 3 registers of 93, 84 and 111bits) total: **288 bits** (36 bytes)
- **Choice** is 4 byte array
- **Choice_Triv[X,Y,Z,W]** => 4 x 36 bytes = 144 bytes

Generation of Pub/private Homomorphic keys with Armadillo/Cingulata:

- Public FHE Key: 762 Kb (duration= ?), Private key: ? Kb (duration= ?)

Crypto conversion token

- FHE token : 762 Kb x 128 bits = 97 536 Kb = 97 Mbytes (duration= ?)

Size of Linux binaries run from Add-on Chrome: 216ko

Multiplicative depth of Targeted advertising Algorithm: 8 probably

11. Annexe A : Stand Alone RSA version of Ad Server System :

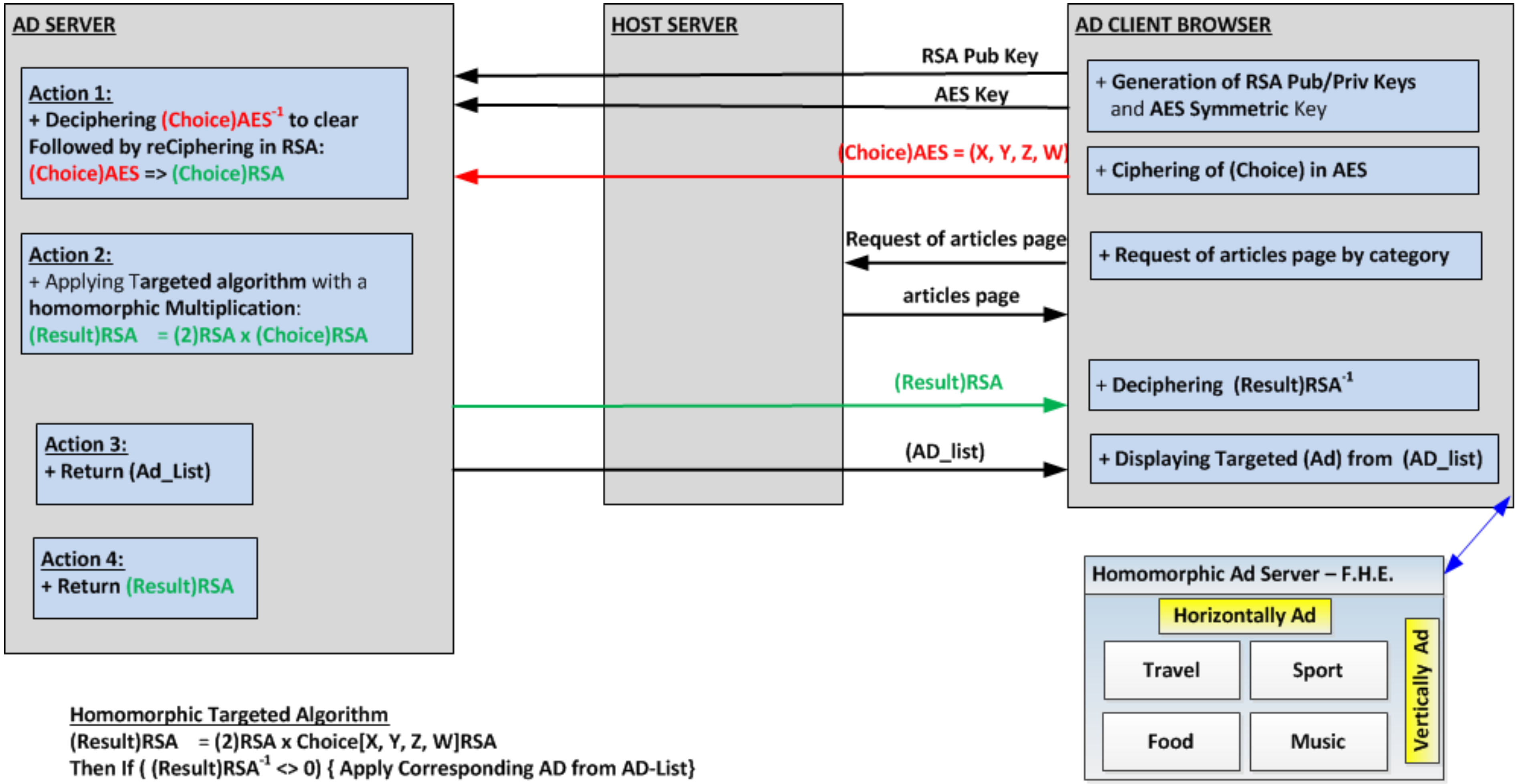
Work done before integrating of CEA libraries:

An important part of the work done for the project was to design and develop a complete Client/Servers architecture in **Node-JS / JavaScript** with a stand alone version of the Cryptography in RSA, which is in fact not Full homomorphic but already Simply Homomorphic.

This allowed us **to design and develop Node-js route of Ad Server and Host Server** and also to design the **Front end of our Cryptocomp Website magazine**.

We are close to finish the version and expect to start trying to build binaries with Cingulata Compiler toolchain in the next days.

11. Annexe B : Architecture diagram of RSA Ad Server



AD SERVER
Action 1:
+ Deciphering (Choice)AES⁻¹ to clear
Followed by reCiphering in RSA:
(Choice)AES => (Choice)RSA
Action 2:
+ Applying Targeted algorithm with a
homomorphic Multiplication:
(Result)RSA = (2)RSA x (Choice)RSA
Action 3:
+ Return (Ad_List)
Action 4:
+ Return (Result)RSA
HOST SERVER
AD CLIENT BROWSER
RSA Pub Key
AES Key
(Choice)AES = (X, Y, Z, W)
Request of articles page
articles page
(Result)RSA
(AD_list)
+ Generation of RSA Pub/Priv Keys and AES Symmetric Key
+ Ciphering of (Choice) in AES
+ Request of articles page by category
+ Deciphering (Result)RSA⁻¹
+ Displaying Targeted (Ad) from (AD_list)
Homomorphic Ad Server – F.H.E.
Horizontally Ad
Vertically Ad
Travel
Sport
Food
Music

Homomorphic Targeted Algorithm

$(Result)RSA = (2)RSA \times Choice[X, Y, Z, W]RSA$

Then If ($(Result)RSA^{-1}$ <> 0) { Apply Corresponding AD from AD-List}

11. Annexe C : Definitions

Bootstrapping: A technic which allows to evaluate arbitrary circuits by essentially evaluating the decryption function on encrypted secret keys and reduce the noise in the previous homomorphic calculation. [**REF-5]**

Somewhat Homomorphic Encryption (SWHE): This scheme enables a homomorphic calculation with a limited noise by choosing a crypto system with a multiplicative depth small and known. [**REF-1]**

BGV scheme: a SWHE scheme which is implemented in the HElib library.

BGV : Brakerski, Gentry, and Vaikuntanathan SWHE scheme.

FV : Fan and Vercauteren SWHE scheme.

Multiplicative Depth of boolean circuit: number of consecutive multiplications used during an homomorphic calculation.

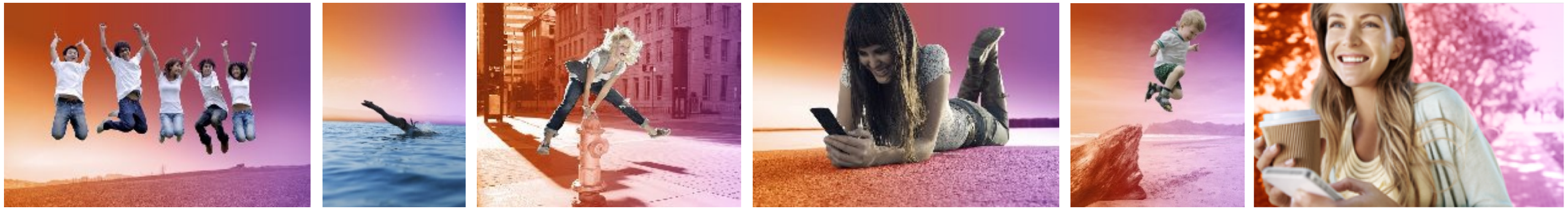
Cingulata Compiler toolchain: Toolchain which allows to transform **CPP input code** into equivalent NAND and XOR **Boolean circuit representation**, and optimize it with **ABC** library. The optimized code is run finally for doing Homomorphic calculation depending on different possible **FHE scheme**. [**REF-1]**

.

11. Annexe D : References

REF-1	Annexe du contrat FUI17 - FICHE TECHNIQUE DE PROJET - SYSTEMATIC PARIS-REGION.
REF-2	Etude et sélection des cas d'usage des démonstrateurs CRYPTOCOMP (version L1.3)
REF-3	CEA-LIST Cingulata Wiki GitHub home page : https://github.com/CEA-LIST/Cingulata/wiki
REF-4	Stream ciphers: A Practical Solution for Efficient Homomorphic-Ciphertext Compression, (from Anne Canteaut, Sergiu Carpov, Caroline Fontaine, Tancrede Lepoint, Marya Naya-Plasencia, Pascal Paillier, Renaud Sirdey. International Conference on Fast Software Encryption 2016: pp 313-333
REF-5	Faster Fully Homomorphic Encryption: Bootstrapping in less than 0.1 Seconds from *Ilaria Chillotti and Nicolas Gama and Mariya Georgieva and Malika Izabachène. 2016*
BGV14	Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) Fully Homomorphic Encryption without Bootstrapping. TOCT, 6(3):13, 2014.

…



End

viaccess·orca