End-to-end Health Data Privacy Using Secure 5G Data Channels DeepSec 2022

Associate Professor Dr. Razvan Bocu

Transilvania University of Brasov, Romania

razvan@bocu.ro

About…

- **Associate Professor (Conferenţiar) in the Department of Mathematics and Computer Science, Transilvania University of Brasov, Romania**
- **Didactic (both BSc and MSc levels) and research duties, which pertain to machine learning and artificial intelligence, complex networks, cloud and distributed infrastructures, software systems engineering, bioinformatics.**

About… (cont'd)

- **Scientific Researcher in the Department of Research and Technology, Siemens Industry Software, Brasov, Romania.**
- **In this capacity, I oversee and I am involved in various research-related activities with strategic and business value.**

Computations on Encrypted Data

The ultimate goal: computations over encrypted data…

… this requires the computation of *both* sums *and* products …

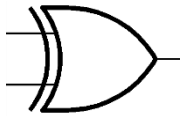
… over the same encrypted data set!

Computations on Encrypted Data

Why SUMs and PRODUCTs?

SUM

=



XOR

0 XOR 0	0
1 XOR 0	1
0 XOR 1	1
1 XOR 1	0

PRODUCT

=

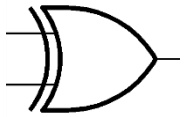
AND

0 AND 0	0
1 AND 0	0
0 AND 1	0
1 AND 1	1

Computations on Encrypted Data

Considering the system {XOR,AND} is Turing-complete …

… any function is a combination of XOR and AND gates



XOR

0 XOR 0	0
1 XOR 0	1
0 XOR 1	1
1 XOR 1	0

AND

0 AND 0	0
1 AND 0	0
0 AND 1	0
1 AND 1	1

Computations on Encrypted Data

Considering the system {XOR,AND} is Turing-complete …

… any function is a combination of XOR and AND gates

Example: Indexing a database

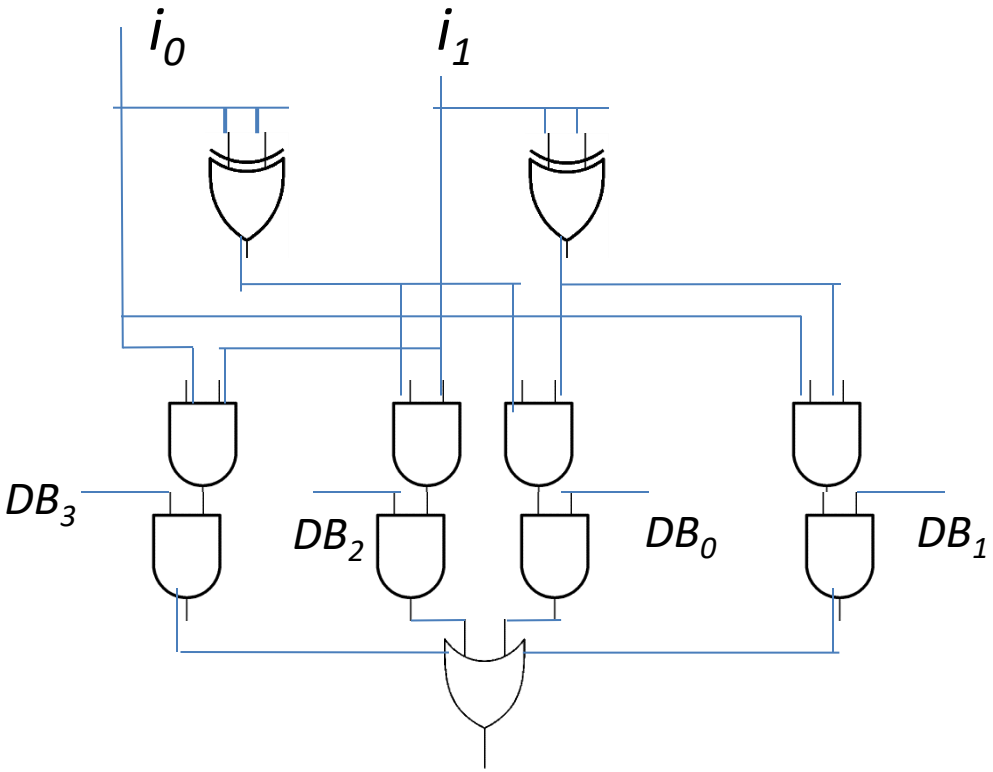
DB

0
1
1
0

index

$i = i_1 i_0$

return DB_i

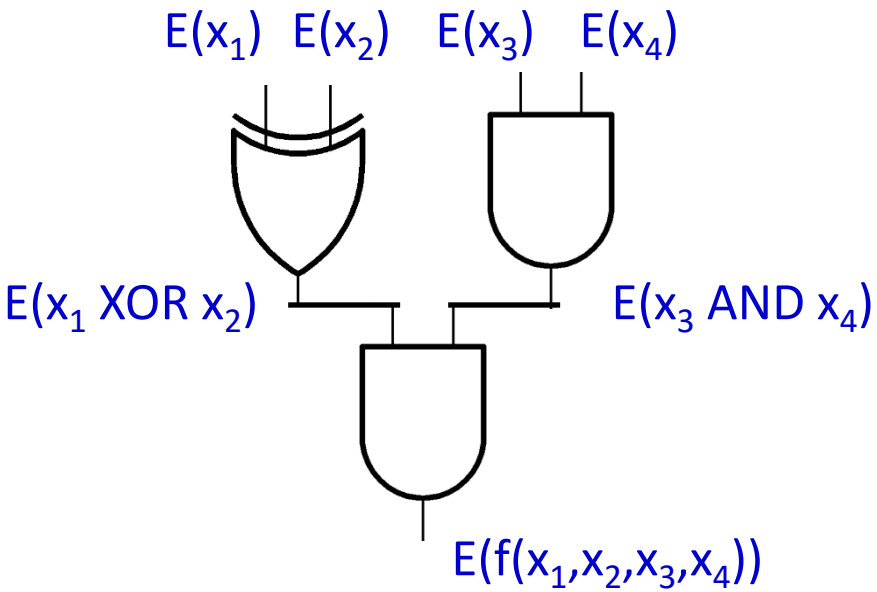


Corollary

Considering the system {XOR,AND} is Turing-complete …

… if one can compute sums and products on encrypted bits

… one can compute ***ANY*** function on encrypted inputs

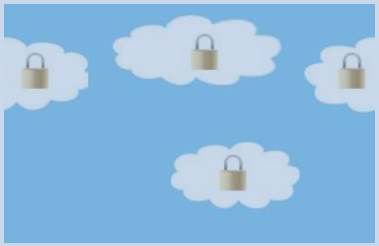


Fully Homomorphic Encryption!

Cryptography's Holy Grail

Fully Homomorphic Encryption!

Applications:



Private Cloud Computing

Microsoft HealthVault

google.com https://mail.google.com/mail/?shva=1#inbox

Delegate *arbitrary processing* of data

without giving away *access* to it

Fully Homomorphic Encryption!

Continuous unsucccessful quest for years

… until, in October 2008 …

… ***Craig Gentry*** came up with the first fully homomorphic encryption scheme …

Forbes.com
U.S. EUROPE ASIA
Home Page for the World's Busines
Home Business Investing Technology
CIO Network Games Gear Green Tech Innovation & Science
Breakthroughs
BM's Blindfolded Calculator
dy Greenberg, 06.24.09, 06:00 PM EDT
rbes Magazine dated July 13, 2009
esearcher's algorithm could teach computers a new
acy trick.

United States [change]
Press room
Services Products Support & downloads My IBM
Press room > Press releases >
IBM Researcher Solves Longstanding Cry
Discovers Method to Fully Process Encrypted
Content; Could Greatly Further Data Privacy and
Computing Security

Slashdot
stories
recent
popular
ask slashdot
book reviews
games
IBM Claims Breakthrough In Analysis of Encr
Posted by timothy on Thursday June 25 2009, @01:13PM
from the scrambled-in-the-shell dept.
An anonymous reader writes
"An IBM researcher has solved a thorny mathematical p
public-key encryption several decades ago. The breakthr
encryption,' makes possible the deep and unlimited analy
scrambled — without sacrificing confidentiality."

Subscribe to a Newsletter
COMPUTERWORLD
Topics News In Depth Reviews Blogs Opinion Shark Tank
Applications
App Development BI and Analytics CRM Content Management
Enterprise Architecture and SOA Enterprise Web 2.0/Collaboration
Virtualization
Cognizant
t done. Click to get started.
Home > Applications > IT Governance and Compliance
News
IBM touts encryption innovation
New technology performs calculations on encrypted data without decrypting it
By Michael Cooney
June 25, 2009 01:03 PM ET
Comments (5) Recommended (26) Like
Network World - IBM today said one of its researchers has made it po
uter systems to perform calculations on encrypted data without
computer services, such as
of others, fully a

What is the mechanism?

What kind of mathematical models can we use?

What kind of objects can we add and multiply?

Polynomials?

$$(x^2 + 6x + 1) + (x^2 - 6x) = (2x^2 + 1)$$

$$(x^2 + 6x + 1) \times (x^2 - 6x) = (x^4 - 35x^2 - 6x)$$

What kind of objects can we add and multiply?

Polynomials?

$$(x^2 + 6x + 1) + (x^2 - 6x) = (2x^2 + 1)$$

$$(x^2 + 6x + 1) \times (x^2 - 6x) = (x^4 - 35x^2 - 6x)$$

Matrices?

$$\begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} + \begin{pmatrix} -1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 3 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} X \begin{pmatrix} -1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} -1 & 1 \\ -1 & 3 \end{pmatrix}$$

What kind of objects can we add and multiply?

Polynomials?

$$(x^2 + 6x + 1) + (x^2 - 6x) = (2x^2 + 1)$$

$$(x^2 + 6x + 1) \times (x^2 - 6x) = (x^4 - 35x^2 - 6x)$$

Matrices?

$$\begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} + \begin{pmatrix} -1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 3 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} X \begin{pmatrix} -1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} -1 & 1 \\ -1 & 3 \end{pmatrix}$$

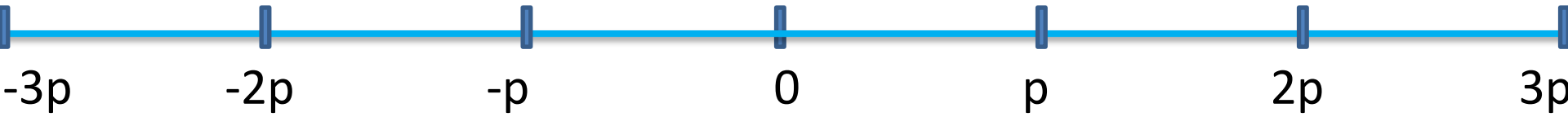
Maybe integers?!?

$$3 + 4 = 7$$

$$3 \times 4 = 12$$

Nowadays, in use: Symmetric Encryption

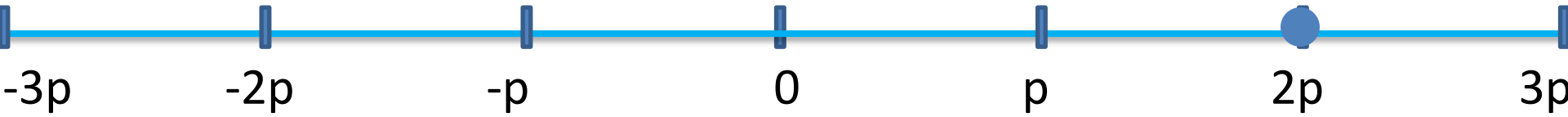
Secret key: large *odd* number **p**



Secret key: large *odd* number **p**

*To Encrypt a bit **b**:*

– choose a (preferably random) “large” multiple of p, say **q·p**



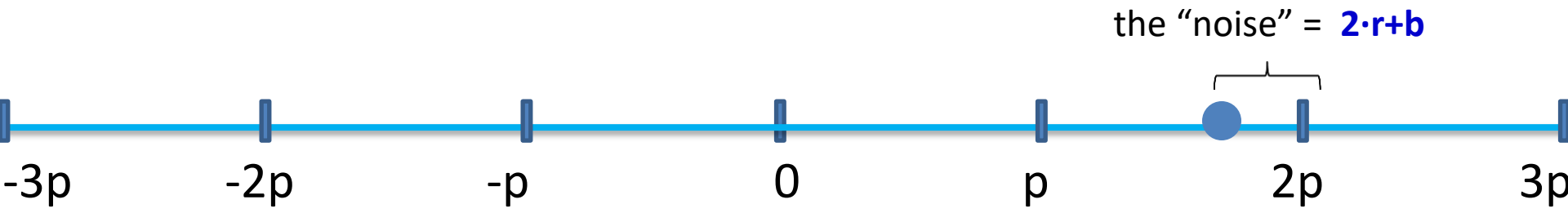
Secret key: large *odd* number **p**

*To Encrypt a bit **b**:*

– choose a (preferably random) "large" multiple of p, say **q·p**

– choose a (preferably random) "small" number **2·r+b**

(this is even if b=0, and odd if b=1)

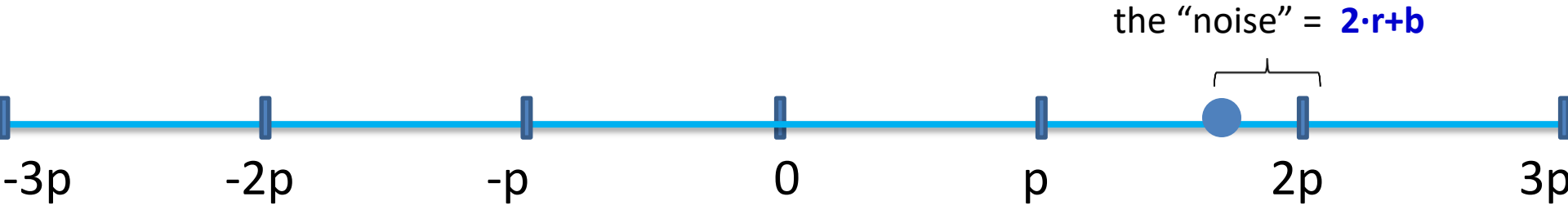


Secret key: large *odd* number **p**

*To Encrypt a bit **b**:*

- choose a (preferably random) "large" multiple of p, say **q·p**
- choose a (preferably random) "small" number **2·r+b**

  (this is even if b=0, and odd if b=1)
- Resulting ciphertext: **c = q·p+2·r+b**



Secret key: large *odd* number **p**

*To Encrypt a bit **b**:*

– choose a (preferably random) "large" multiple of p, say **q·p**

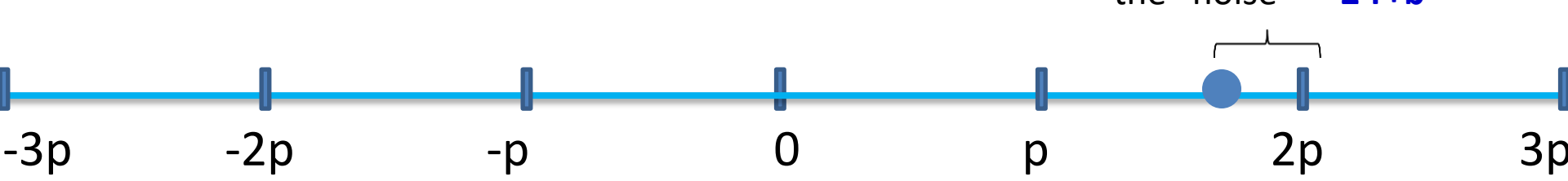
– choose a (preferably random) "small" number **2·r+b**

(this is even if b=0, and odd if b=1)

– Resulting ciphertext: **c = q·p+2·r+b**

*To Decrypt a ciphertext **c**:*

Applying the operation ***c** mod p* recovers the noise



How safe is this model?

If there was no noise (r=0)

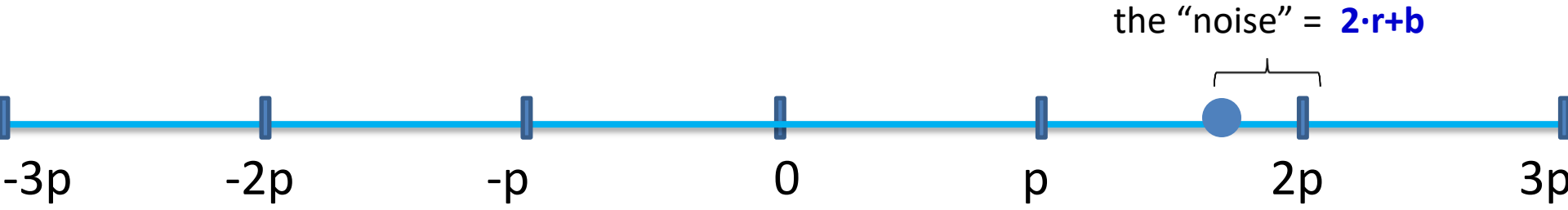
… and one provides two encryptions of 0 (q_1p & q_2p)

… then the secret key p can be recovered

➔ GCD_attack(q_1p, q_2p)

➔ **Greatest common divisor**

➔ **Coppersmith's attack**



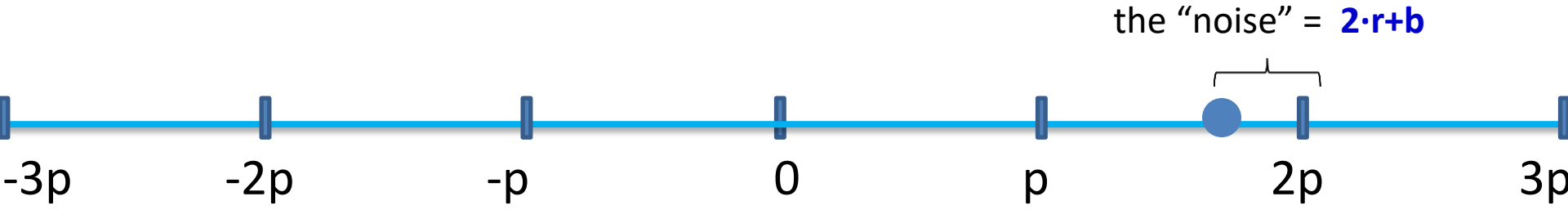
How safe is this model?

If there is noise

… the GCD attack doesn't work

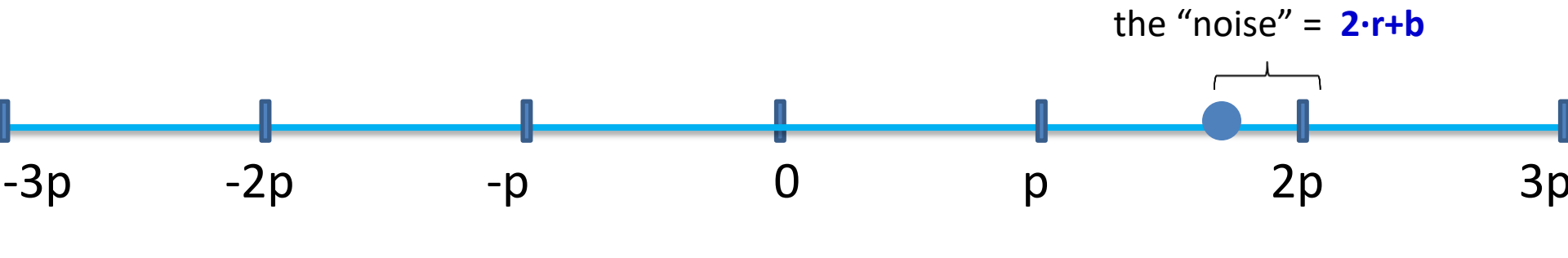
… and neither does any conventional attack

➔ the *approximate GCD assumption*



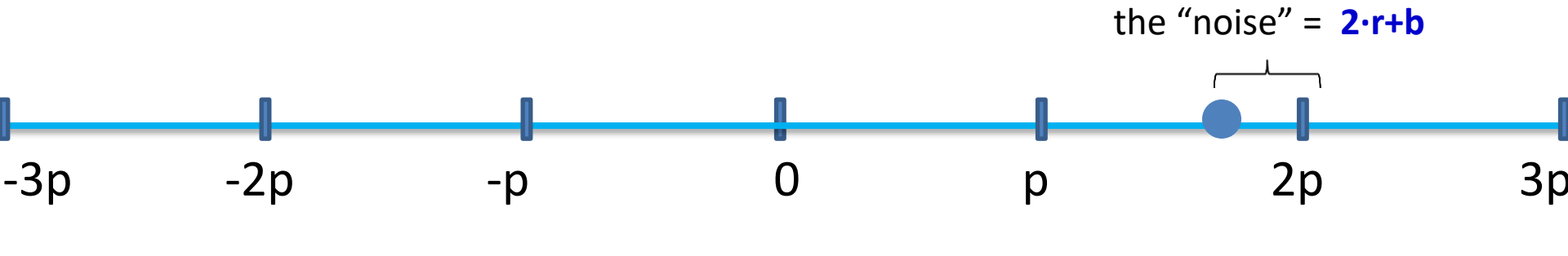
XOR operations on two encrypted bits:

- $\mathbf{c_1} = q_1 \cdot p + (2 \cdot r_1 + b_1)$
- $\mathbf{c_2} = q_2 \cdot p + (2 \cdot r_2 + b_2)$



XOR operations on two encrypted bits:

- $\mathbf{c_1} = q_1 \cdot p + (2 \cdot r_1 + b_1)$
- $\mathbf{c_2} = q_2 \cdot p + (2 \cdot r_2 + b_2)$
- $\mathbf{c_1 + c_2} = \mathbf{p} \cdot (q_1 + q_2) + \mathbf{2} \cdot (r_1 + r_2) + (b_1 + b_2)$

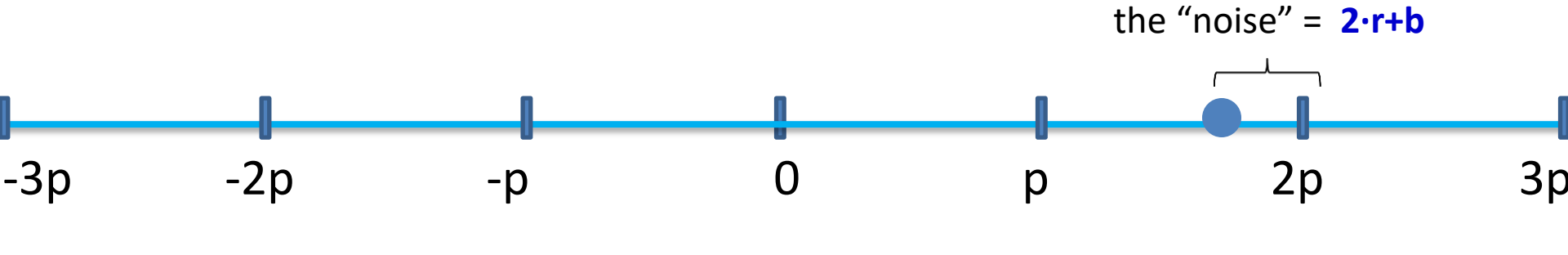


XOR operations on two encrypted bits:

- $\mathbf{c_1} = q_1 \cdot p + (2 \cdot r_1 + b_1)$
- $\mathbf{c_2} = q_2 \cdot p + (2 \cdot r_2 + b_2)$
- $\mathbf{c_1 + c_2} = \mathbf{p} \cdot (q_1 + q_2) + \underbrace{\mathbf{2} \cdot (r_1 + r_2) + (b_1 + b_2)}$

Odd if $b_1=0, b_2=1$ (or) $b_1=1, b_2=0$

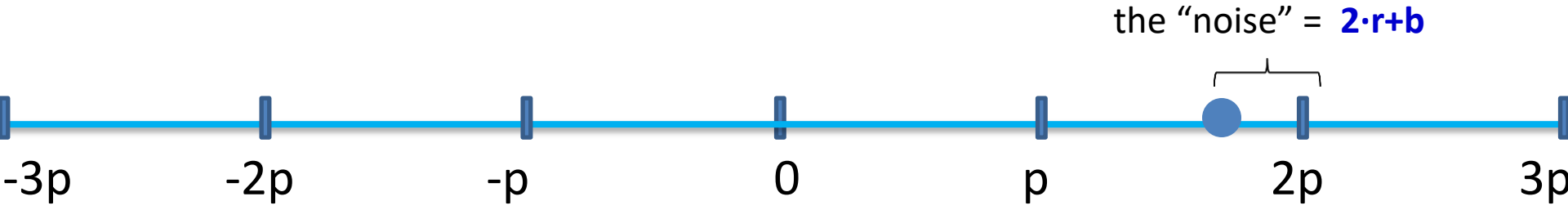
Even if $b_1=0, b_2=0$ (or) $b_1=1, b_2=1$



XOR operations on two encrypted bits:

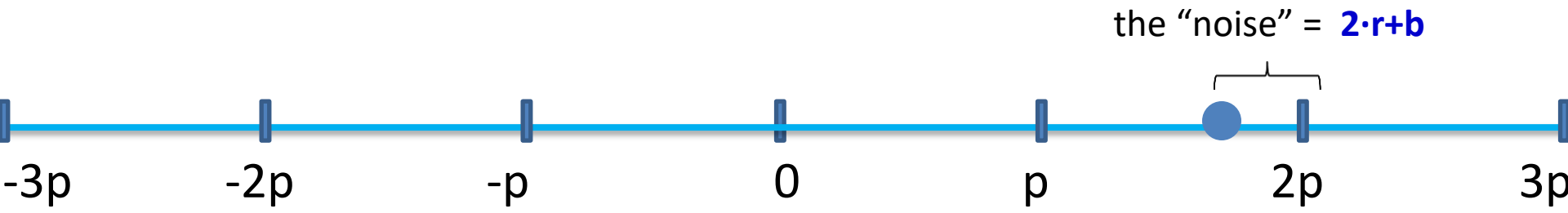
- $\mathbf{c_1} = q_1 \cdot p + (2 \cdot r_1 + b_1)$
- $\mathbf{c_2} = q_2 \cdot p + (2 \cdot r_2 + b_2)$
- $\mathbf{c_1 + c_2} = \mathbf{p} \cdot (q_1 + q_2) + \underbrace{2 \cdot (r_1 + r_2) + (b_1 + b_2)}$

least_significant_bit= b_1 XOR b_2



AND operations on two encrypted bits:

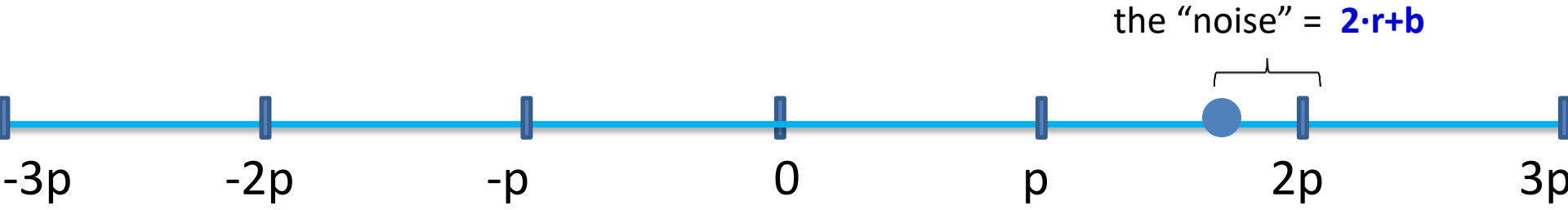
- $\mathbf{c_1} = q_1 \cdot p + (2 \cdot r_1 + b_1)$
- $\mathbf{c_2} = q_2 \cdot p + (2 \cdot r_2 + b_2)$
- $\mathbf{c_1 c_2} = \mathbf{p} \cdot (c_2 \cdot q_1 + c_1 \cdot q_2 - q_1 \cdot q_2) + \mathbf{2} \cdot (r_1 r_2 + r_1 b_2 + r_2 b_1) + b_1 b_2$



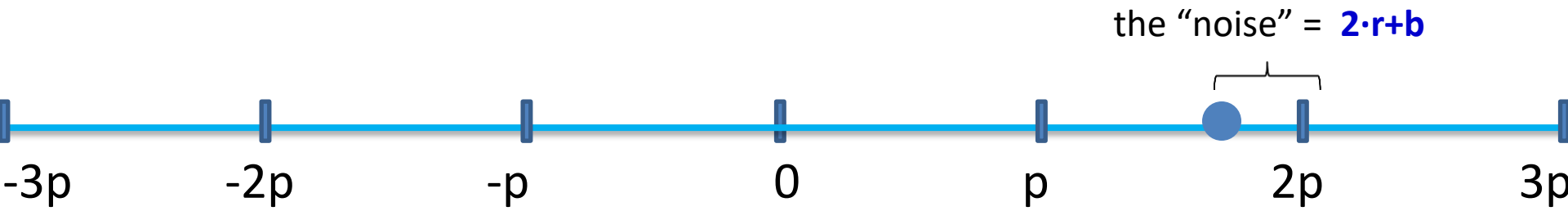
AND operations on two encrypted bits:

- $\mathbf{c_1} = q_1 \cdot p + (2 \cdot r_1 + b_1)$
- $\mathbf{c_2} = q_2 \cdot p + (2 \cdot r_2 + b_2)$
- $\mathbf{c_1 c_2} = \mathbf{p} \cdot (c_2 \cdot q_1 + c_1 \cdot q_2 - q_1 \cdot q_2) + \underbrace{2 \cdot (r_1 r_2 + r_1 b_2 + r_2 b_1) + b_1 b_2}$

least_significant_bit= b_1 AND b_2



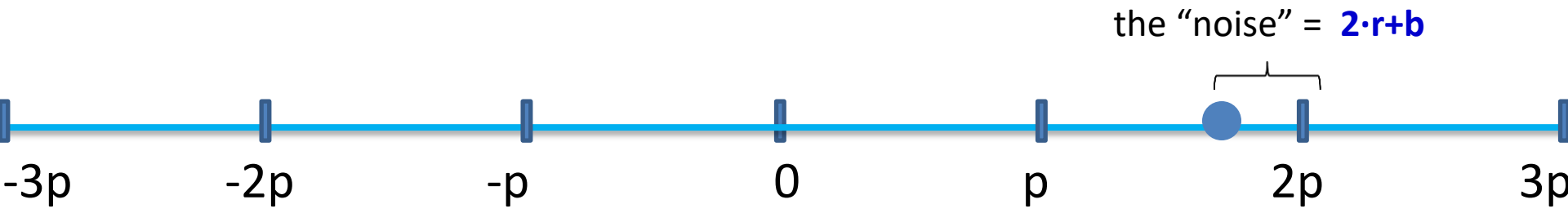
The noise increases!



The noise increases!

$$-\mathbf{c_1+c_2} = \mathbf{p}\cdot(q_1 + q_2) + \underbrace{\mathbf{2}\cdot(r_1+r_2) + (b_1+b_2)}$$

noise= 2 * (initial noise)



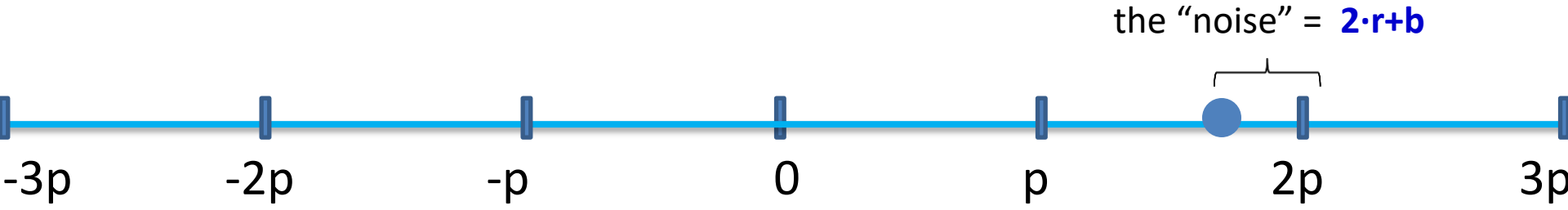
The noise increases!

$$-\mathbf{c_1+c_2} = \mathbf{p}\cdot(q_1 + q_2) + \underbrace{\mathbf{2}\cdot(r_1+r_2) + (b_1+b_2)}$$

noise= 2 * (initial noise)

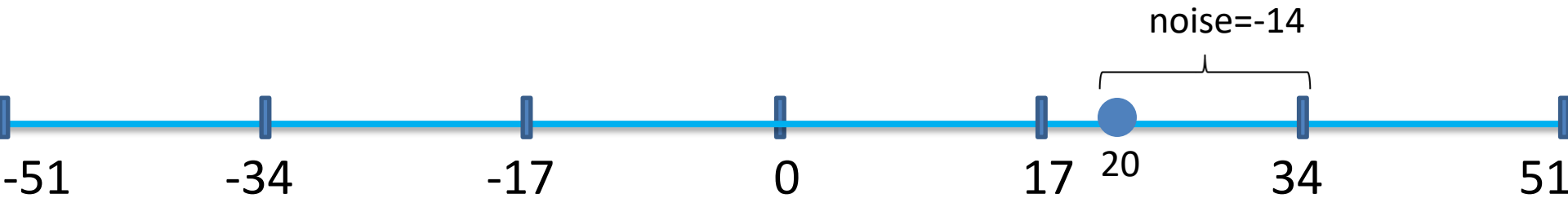
$$-\mathbf{c_1c_2} = \mathbf{p}\cdot(c_2\cdot q_1+c_1\cdot q_2-q_1\cdot q_2) + \underbrace{\mathbf{2}\cdot(r_1r_2+r_1b_2+r_2b_1) + b_1b_2}$$

noise = (initial noise)^2



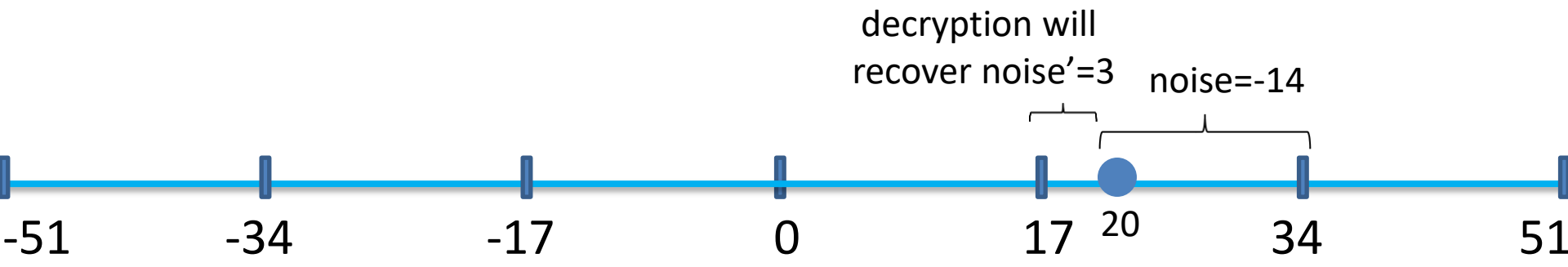
The noise increases!

Why does this matter?



The noise increases!

Why does this matter?

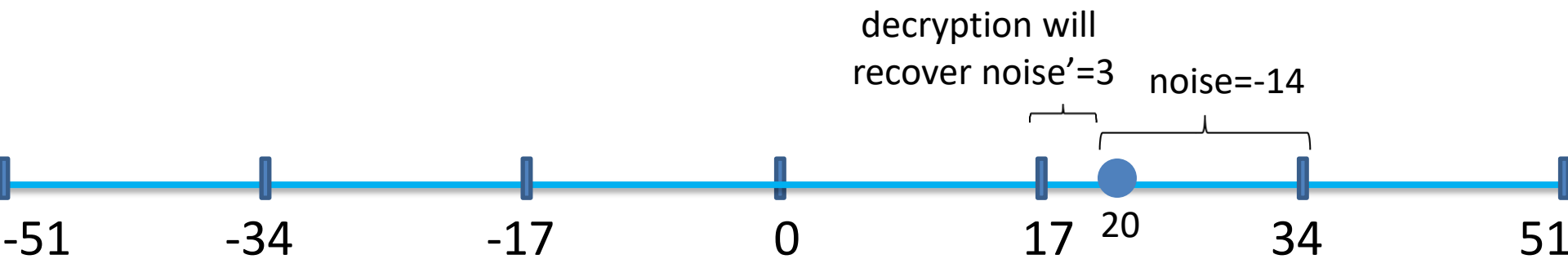


The noise increases!

Why does this matter?

If the |noise| > p/2, then:

Decryption will output an incorrect bit!



The accomplishment …

Possibility to do lots of additions and

… some multiplications

(= a “somewhat homomorphic” encryption)

The accomplishment …

… we can do lots of additions and

… some multiplications

(= a "somewhat homomorphic" encryption)

It is enough to do many useful tasks, such as, database search, spam filtering etc.

The accomplishment …

… we can do lots of additions and

… some multiplications

(= a "somewhat homomorphic" encryption)

… enough to do many useful tasks, e.g.,
database search, spam filtering etc.

But, there is much more …

RSA&friends

Fully homomorphic

MANY mult

ZERO add

WE ARE HERE!

MANY additions

MANY multiplications

The *"bootstrapping method"*

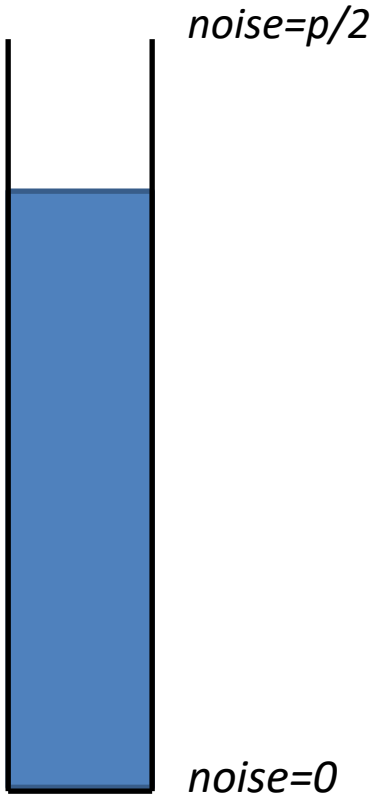
Principle: If you can go a (large) part of the way, then you can go all the way.

How is this possible?

The *"bootstrapping method"*

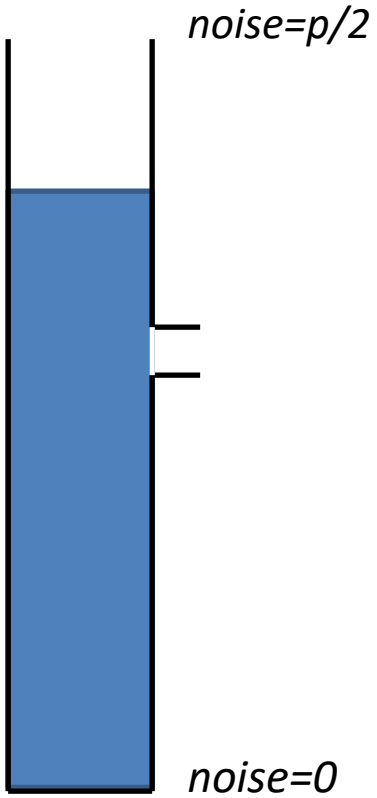
Initial noise

The *"bootstrapping method"*



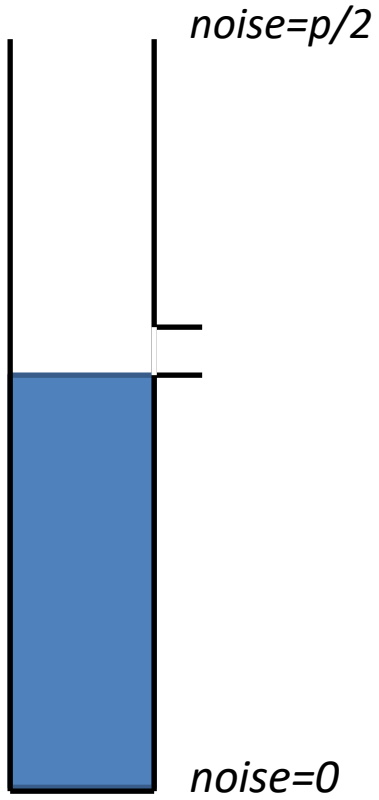
Noise after some sums and products

The "bootstrapping method"



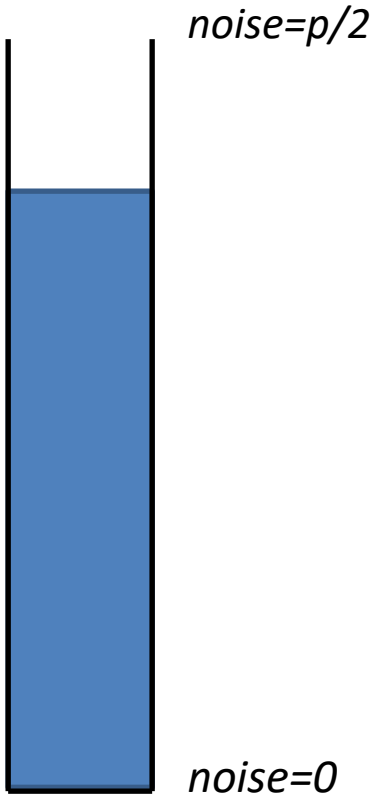
Bootstrapping =
"Valve" at a fixed height

The "bootstrapping method"



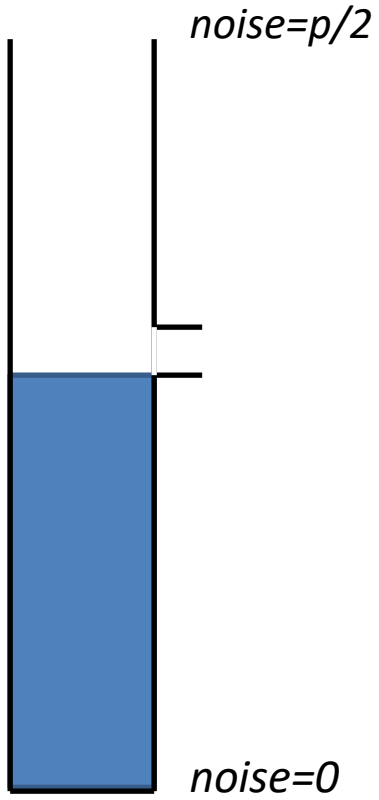
Bootstrapping =
"Valve" at a fixed height

The "bootstrapping method"



… repeat until done

The *"bootstrapping method"*

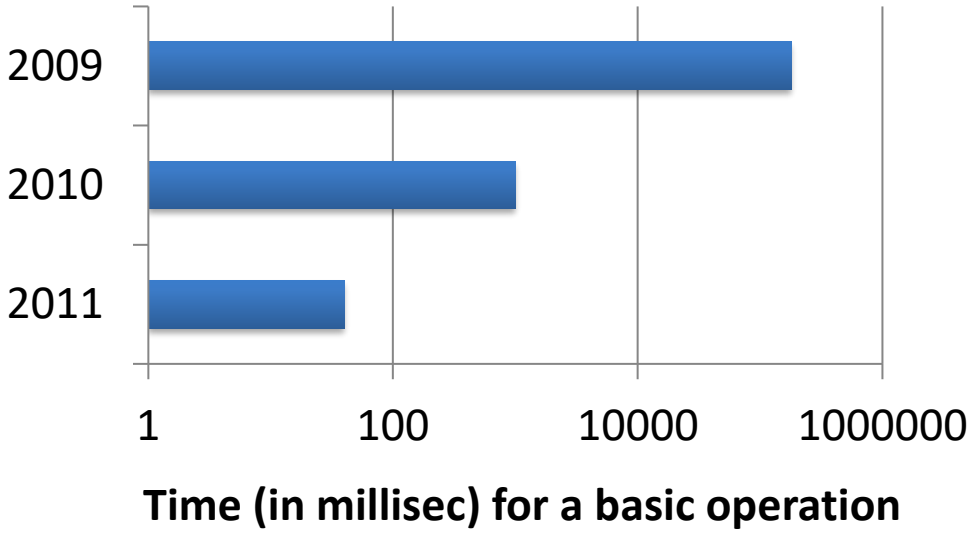


… repeat until done

- *Lots of new Encryption Schemes*

  … simpler, more secure, more efficient

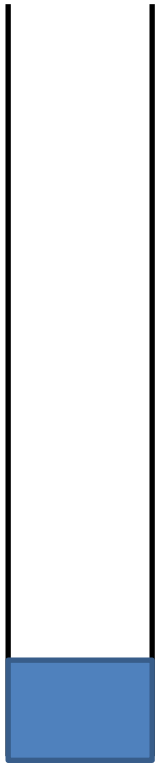
- *Dramatic Efficiency Improvements*



*Gentry's **"bootstrapping method" …***

The same principle: if you can go a (large) part of the way, you probably can go all the way.

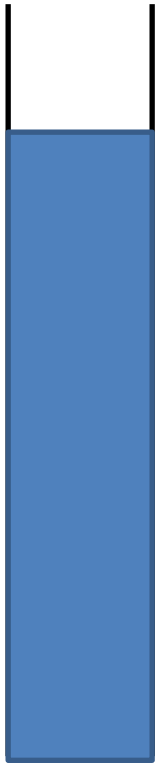
noise=p/2



noise=0

*Gentry's **"bootstrapping method" …***

The same principle: if you can go a (large) part of the way, you probably can go all the way.



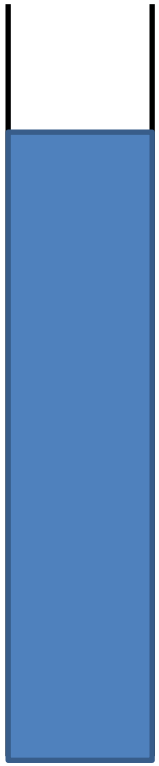
Issue to address: Addition and Multiplication increase noise

(Addition doubles, Multiplication squares the noise)

*Gentry's **"bootstrapping method" …***

The same principle: if you can go a (large) part of the way, you probably can go all the way.

noise=p/2



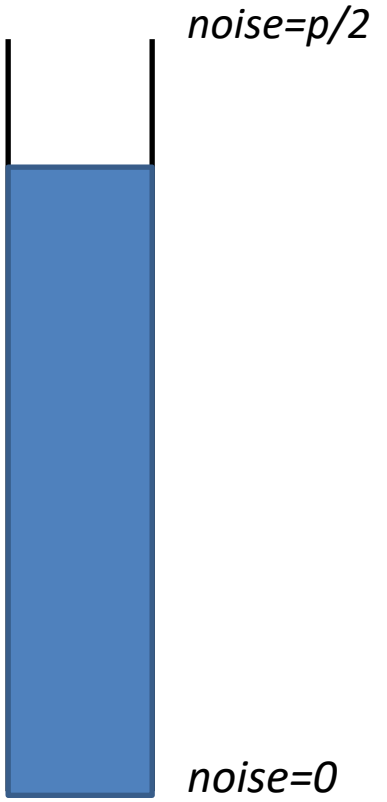
noise=0

Issue to address: Addition and Multiplication increase noise (Addition doubles, Multiplication squares the noise)

Goal: *noise reduction*

Reflection topic

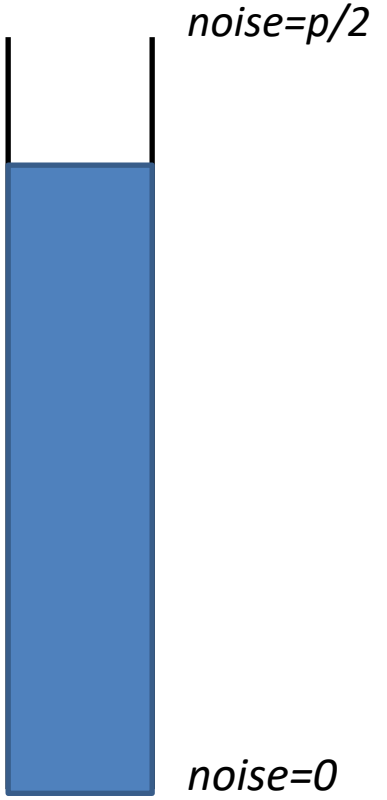
What is the best noise-reduction procedure?



Reflection topic

What is the best noise-reduction procedure?

… To get rid of all the noise.

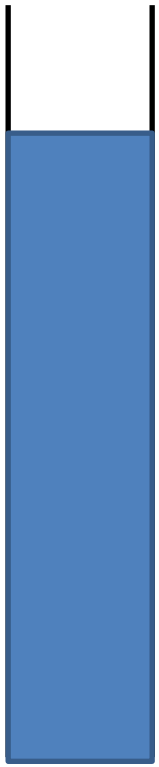


Reflection topic

What is the best noise-reduction procedure?

… To get rid of all the noise,

… and computationally optimal recover the original message.



Real World Solution

Reference contributions:

- Razvan Bocu, Cosmin Costache, **A Secure Distributed e-Health System for the Management of Personal Health Metrics Data, *IBM Journal of Research and Development* 62(1): 1:1-1:10 (2018).**
- Razvan Bocu, Anca Vasilescu, Delia Monica Duca Iliescu, **Personal Health Metrics Data Management Using Symmetric 5G Data Channels, *Symmetry* (2022), 14(7), 1387.**
- Use case: the convenient and full privacy preserving collection, transportation, processing, analysis, and storage of personal health information (PHI) using virtualized and secured 5G data channels.
- e-Health System – this system addresses the four essential requirements, the biomedical data collection at the user's end, its transfer to the storage and processing backend, the proper and secure storage of this data, and its privacy-preserving processing.
- Distinctive feature: clear separation between the long-term data storage and data processing paths. The system can easily accommodate any use case that involves the data collection through sensors and mobile devices at the user's side.

Fundamental Requirements

- The biomedical data is gathered at the user's end.
- It is transferred to the storage and processing backend using virtualized 5G data channels.
- There, it is properly and securely stored.
- The data is processed while completely preserving the privacy of the personal data.

Distinctive Features

- The e-Health system is one of the few personal health information collection frameworks that combine the clear separation between the long-term data storage and data processing paths with the capability to easily attach a variety of medical sensors and data collection devices at the client side.

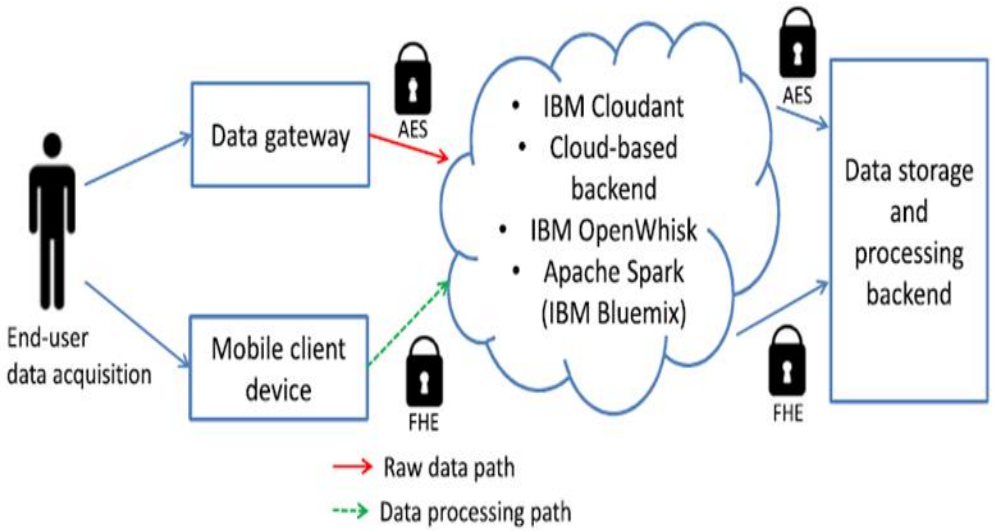
Distinctive Features (2)

- The backend component is able to make use of cloud storage and processing services, which ensure the system's scalability in the future.
- Consequence: the system's functional and architectural features essentially make it the only, as far as we reasonably know, existing distributed system of this kind that processes large amounts of personal medical data in a timely manner, while completely preserving the data privacy.

Stages of data processing

- The first stage pertains to the data acquisition using each enrolled individual's wearable or mobile device.
- The second stage relates to the safe data transmission to the backend components using the secure 5G data channels built using Ericsson NFVI (Network Function Virtualization Infrastructure).
- The third stage pertains to the actual storage of the collected personal health information data.
- The last stage implements the privacy-preserving data processing.
- Consequence: the system's functional and architectural features essentially make it the only, as far as we reasonably know, existing distributed system of this kind that processes large amounts of personal medical data in a timely manner, while completely preserving the data privacy through virtualized 5G data channels.

System Architecture



System Features

- Data privacy assured during all four stages: data collection, data transmission using Ericsson NFVI virtualized 5G , data storage, FHE-based data processing.
- Data storage and processing backend is deployed in the cloud (in this case, IBM Bluemix, but any other cloud platform is fine).
- The collected data is efficiently store in the cloud (in this case, the relevant service is IBM Cloudant, but any other similar cloud service is fine).
- The FHE computations are performed using Apache Spark, but any other computing service may be adapted and used.
- The processing events are intercepted, and the proper actions triggered using a programming service (in this case, IBM OpenWhisk, but any other similar service may be adapted).
- Advantages
  - Any use case that involves the safe (private) processing of sensitive data can benefit from the usage of this model.
  - The approach offloads the expensive processing operations to the cloud infrastructure, while keeping intact the data privacy.
  - The model is fully customizable and adaptable to various use cases and hardware/software infrastructures.

Remarks concerning the 5G NFV

- The virtualized wireless network function (VWNF) is fundamental for the efficient design and implementation of 5G networks. This has been considered in order to deploy the core of the virtual 5G network that supports the function of the integrated data management system. This approach allows for the self-sufficient specification of the logical 5G network, which supports the overall operation of the integrated data management system. Moreover, it also creates the possibility for the specialized logical 5G network that supports the function of the integrated data management system to be deployed on certain hardware and software infrastructures, such as those that are offered by cloud service providers or telecommunications service providers. This mechanism was used in order to specify and deploy the necessary specialized networked services. We have observed that the virtualized networked environment supplies the necessary logical flexibility and scalability.

Remarks concerning the 5G NFV

- This logical mechanism is compatible with the proper processing of the personal health information data traffic that is sent through the logically defined 5G network.
- This approach optimizes the allocation and usage of the necessary radio resources. Thus, we were able to define logical sub-networks that are conducting distinct analyses of the 5G data traffic using individual instances of the integrated data management system. It can further be stated that the experimental work that we conducted acknowledges that logical 5G networks that are adequately defined and sized are capable to support even information systems that work with large amounts of real-time data, such as the integrated data management system.

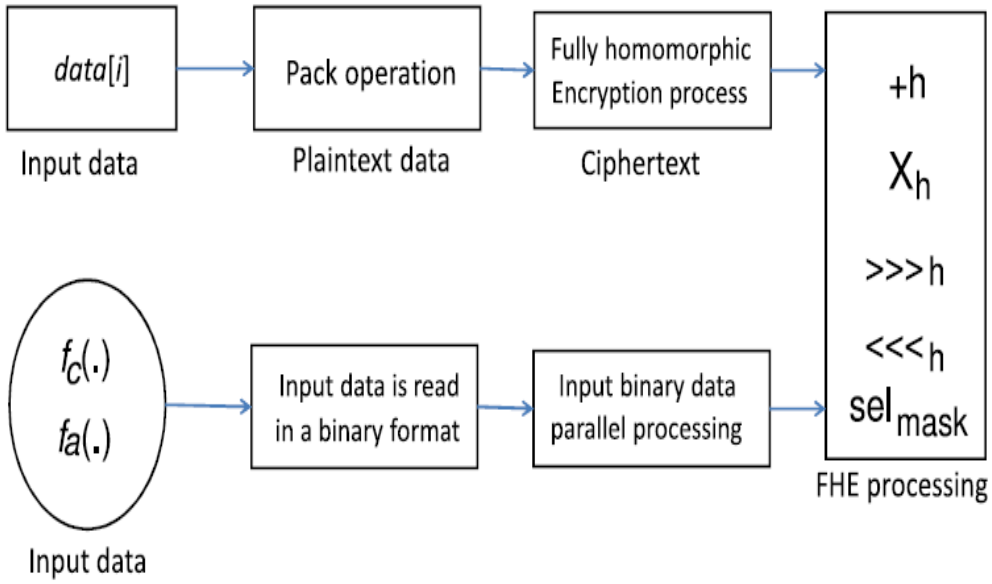
FHE Core Model – Supported Operations

- Homomorphic addition ($+_h$) – It takes as operands two ciphertexts, which correspond to a slot wise XOR operation of the related plaintext elements.
- Homomorphic multiplication (X_h) – It takes as operands two ciphertexts, which correspond to a slot wise AND operation of the related plaintext elements.
- Homomorphic rotate ($<<<_h$, $>>>_h$) – This essentially provides the possibility to rotate the data elements' slots. The concept of *slots* refers to the storage bits that determine the data elements processed by the rotate operation.
- Homomorphic select (sel_{mask}) – It has the role to correct the potentially altered slots (bits) of the data elements after the rotate operation. It preserves the data consistency during the fully homomorphic encryption process.

FHE Core Model – The Level

- The level (L) – It must be determined before starting any computation instruction.
- The level *L* is calibrated considering the depth of the multiplication operations to be performed in the given computational context.
- This parameter assures the accuracy of the FHE operations' results.
- The multiplication increments by 1 the level L of the operation.
- The depth of the multiplication operations determines the value of the calibrated level L.
- This operation considers a number of N_{CT} ciphertexts, which encrypt an array with *n* bits that stores the relevant data (in the case of the e-Health system, the cardiac rhythm data).
- The computationally expensive multiplication operations should be reduced.
- Consequently, the depth of the multiplication operations is reduced, in order to achieve an optimal calibration of the level L.

Optimized FHE Scheme



Optimized FHE Scheme (cont'd)

- The data storage and processing backend efficiently and safely computes the received data.
- The efficient incorporation of the FHE routines into the e-Health system relies on the utilization of the communication data path illustrated in the previous slide (the top data path).
- Each bit of the plaintext data is properly packed into the respective plaintext message.
- The ciphertext is generated through an FHE model considering the top data path steps, which is based on the NTRU encryption algorithm.
- The bottom data path in the figure implies that the input data is translated into a binary format, which is efficiently understood by the CPU. This is achieved using the computation (*fc*(.)) and aggregation (*fa*(.)) functions from the bottom data processing path.
- The binary data is processed using a parallel single instruction, multiple data (SIMD) model.
- The four operations already mentioned are fully supported.

NTRU Encryption Algorithm – Why?

- Lattice-based alternative to RSA and elliptic-curve cryptography.
- It is based on the shortest vector problem in a lattice.
- This is known as immune to quantum computer-based attacks.
- It relies on the presumed difficulty of factoring certain polynomials in a truncated polynomial ring into a quotient of two polynomials having very small coefficients.
- It is more time efficient than other existing approaches.

e-Health System

- The efficient incorporation of the NTRU-based improved fully homomorphic encryption primitives into the e-Health system relies on the utilization of the communication data path.
- The system's architecture is sufficiently flexible in order to address any use case scenario that implies the client data collection through a mobile device, and its safe transportation, storage and processing at the backend. Therefore, it is suitable for the implementation of the smart city medical data system, which is presented in this paper.
- The system is configurable, and it can accommodate existing and future mobile devices, which gather the health data at the user's end.

e-Health System (2)

- The speedup that is induced by the improved NTRU-based data processing module is demonstrated using the relevant metrics.

Performance Factors

- The value of the level *L*, which determines the operation of the FHE model, is an important performance factor.
- The system performance is related to the number of multiplication and rotation operations, which are computationally expensive, and influence the calibration of the level L.
- The computational core of the e-Health system considers the reduction of the level *L*, and also ensures that the number of FHE operations is kept at a minimum.

Performance Factors (2)

- The optimal value of the level *L* is permanently computed.
- This operation considers a number of N_{CT} ciphertexts, which have the role to encrypt an array with *n* bits that stores cardiac rhythm data.
- The improved computational behaviour of the model is also related to the inclusion of the NTRU-based fully homomorphic encryption capabilities into the core of the system.

Medical Conditions

- The detection of three medical conditions has been considered: the average heart rate, the delayed repolarization of the heart, the minimum and maximum heart rates.
- Outcomes:
  - The model performed well considering the detection of all three medical conditions.
  - The resulted performance metrics prove that the system is time and resources efficient.
  - The data privacy can be preserved, even if the hosting (cloud) environment is affected by a security incident (e.g., unauthorized access by an employee or hacker, CPU vulnerability issues, etc.).
  - The amount of transferred data depends arithmetically on the size of the encrypted data.

The Average Heart Rate

- The computation of the average heart rate is based on the storage of the encrypted values in N_{CT} ciphertexts. The implementation optimization that is included in the e-Health system considers three main types of improvements.
- The first one pertains to the reduction of the computationally expensive multiplication operations.
- The second one considers the reduction of the computation operations depth, so that the level *L* is calibrated at the optimal level.
- Finally, the implementation of the NTRU-based encryption routines further improves the efficient usage of the computational resources, including the processor time allocation.

The Average Heart Rate (2)

- The addition operation is optimized considering two mechanisms: the *additive compression* and the *prefixed parallel addition*.
- The additive compression transforms three data inputs *(H,M,F)*, each of them composed of *n* bits, into two outputs. These are represented by the A_{R} (addition result), and L_{OVER} (leftover). Thus, the $A_{\mathrm{R}} = H\Delta M\Delta F$, and $L_{\mathrm{OVER}} = [(H \times M)\nabla(H \times F)\nabla(M \times F)] \ll 1$. Here, Δ denotes an additive single instruction multiple data (SIMD) operation, while the *nabla* operand (∇) also represents a SIMD operation, which is conducted on all *n* bits of the input data in a parallel manner.

The Average Heart Rate (3)

- The computation of the average heart rate considers the N_{CT} ciphertexts, which encrypt the input messages that are represented on n bits.
- The first step of this data flow relies on the usage of the additive compression in order to transform N_{CT} ciphertexts into two ciphertexts.
- The resulting two ciphertexts are added using the prefixed parallel addition operation.

The Detection of DRHS

The fundamental equation:

$$T_{QT}/\sqrt{T_{RR}} > 475\ \text{ms} \Rightarrow {T_{\text{QT}}}^2 > T_{\text{RR}} \times 225{,}625\ (1)$$

$$\Rightarrow T_{\text{QTH}} > T_{\text{RRH}} (2)$$

The Detection of DRHS (2)

- The expressions ${T_{\mathrm{QT}}}^2 = T_{\mathrm{QTH}}$ and $T_{\mathrm{RR}} \times 225{,}625 = T_{\mathrm{RRH}}$, are computed using the frontend, client-side devices that are represented in Figure 1. The T_{QT} and T_{RR} represent the time intervals that are measured and recorded during any electrocardiogram test. In principle, T_{QT} represents the time taken for ventricular depolarisation and repolarisation, while T_{RR} measures the variability in the timing of the heartbeats. The subscript *H* denotes the homomorphic nature of the comparison, which detects the existence of the DRHS condition. The optimized version of the equation (1), which has been obtained after an extensive set of calibration tests, has been considered during the implementation of the e-Health system.

The Detection of DRHS (3)

- The equation is optimized regarding the accuracy of the detection, and the efficient usage of the computational resources.
- The equation ensures that the e-Health system accurately detects the DRHS condition with virtually no false positives, while running only the absolute necessary NTRU-based FHE operations.
- The data storage and processing backend aggregates the results of the individual comparisons.
- The client device requests a report from the backend considering a certain period of time.
- The client device decrypts the received result and checks for the presence of at least one bit that is equal to 1.
- If yes, then during the given time period the comparison $T_{\mathrm{QTH}} > T_{\mathrm{RRH}}$ was true at least once. Consequently, the DRHS condition occurred with a significant probability at least once.

The Detection of the Minimum and the Maximum Heart Rates

- Functional requirement of the system and is implemented considering the $f(.)$ function.
- The function is intended to convert the input data into a binary format, which is efficiently processed by the system.
- The comparison of two numbers, which are defined by *n* bits, produces a result that is also determined by *n* bits.
- If the first number is greater than the other number then the result will contain a single bit of 1, and *n-1* bits that have a value of 0.
- If the first number is less than the other number, then the result contains only bits with a value of 0.
- The e-Health system triggers a succession of rotate and select operations. The output of this subroutine is represented by a succession of *n* bits, each with a value of 1.

The Detection of the Minimum and the Maximum Heart Rates (2)

- The problem of determining the minimum and the maximum values for the cardiac rate is reduced to the problem of determining the minimum and the maximum values from among N_{CT} ciphertexts, which encrypt an array of messages that are composed of *n* bits.
- Thus, the proper calculation of the minimum and maximum values for the cardiac rate is based on the successive application of the following functions: min($f_c(.)$) and max($f_c(.)$). In this context, the initial calibrated level *L* of the NTRU-based fully homomorphic encryption computation is calculated according to the following reference formula: $L > (\log_2 n + 2) \times \log_2 N_{\mathrm{CT}}$.

Remarks

- One of the system's important features is its ability to accommodate any kind of user-side (client-side) mobile data collection device, provided that it is technically capable of gathering the required personal health information.
- The structural versatility and stability of the system is determined by the fact that only the user-side data collection devices may vary.

Remarks (2)

- Any technically suitable user-side device is able to communicate with the system and send the data to the data storage and processing backend, without any hardware topology changes.
- The client software module, which is installed on the user's mobile device, is able to send the collected data to the backend in real time.
- If the data connection is not available, then the collected data is stored locally, and immediately transferred to the backend as soon as a working data connection becomes available.

Study Population and Hardware

- 750 citizens of Brasov City, Romania.
- We have tested a variety of personal cardiac rate sensors, and determined that the most accurate device is the Polar H10.
- Thus, it has been decided to use this device in order to gather the cardiac rate data, which is necessary to assess the system's ability to detect the delayed repolarization of the heart syndrome (DRHS).

Data Pathway

- The cardiac rate data is gathered by the Polar H10 personal sensor.
- The collected data is sent to each person's Android smartphone.
- The e-Health system's client application is installed on the smartphone.
- It collects the data, which is transmitted by the personal sensor, properly encrypts it and sends it to the data storage and processing backend, which is stored inside the IBM Bluemix infrastructure.

Performance Metrics - Explanation

- Network capacity: $XFER_{IN}$ (the amount of data transferred from the client devices to the backend), $XFER_{OUT}$ (the amount of data that is transferred from the backend to the client devices).
- Storage ratio (S_R): this assesses the amount of storage that is necessary to store one byte of plaintext data in a FHE format. As an example, if S_R=500, there are necessary 500 bytes in order to store one plaintext byte in the FHE format.
- Processing speed (P_S): This is defined through the ratio $P_S=P_{TO} / P_{IN}$. Here, the numerator represents the amount of time to send the data from the client device to the backend, while the denominator is the amount of time that is required by the backend to process the received data.
- N_{CT}: The number of the involved ciphertexts.
- Level L: The value of the calibration parameter.
- L_{5G}: the load on the virtualized 5G data channel.

Performance Metrics Values

Data Reading Interval	L_{5G}	N_{CT}	**Level L**	$XFER_{IN}$	$XFER_{OUT}$	S_R	P_S
One minute	0.01	2	10	5.3	3201.3	32.1	0.54
Five minutes	0.07	12	12	6.4	1298.8	39.4	0.24
Fifteen minutes	0.19	40	15	6.9	669.2	47.5	0.23
Thirty minutes	0.33	44	16	10.6	1102.6	88.3	0.36
One hour	0.39	86	18	8.1	643.7	91.4	0.35
Three hours	0.52	258	20	9.6	221.9	101.2	0.37
Six hours	0.63	519	21	11.6	108.8	108.5	0.36
Twelve hours	0.72	1021	23	12.1	45.9	117.4	0.39
One day	0.81	2099	25	15.2	26.4	128.1	0.42

Performance Metrics Values – DRHS Condition

Data Reading Interval	L_{5G}	N_{CT}	Level L	$XFER_{IN}$	$XFER_{OUT}$	S_R	P_S
One minute	0.02	2	11	6.42	4104.3	32.1	0.79
Five minutes	0.09	12	14	9.47	1681.8	39.4	0.41
Fifteen minutes	0.22	40	18	8.1	865.2	47.5	0.32
Thirty minutes	0.36	44	21	11.79	1602.9	88.3	0.43
One hour	0.42	86	24	9.85	814.8	91.4	0.41
Three hours	0.55	258	27	10.87	314.8	101.2	0.44
Six hours	0.67	519	31	12.9	198.9	108.5	0.39
Twelve hours	0.76	1021	35	13.84	87.9	117.4	0.45
One day	0.86	2099	39	16.86	26.4	208.1	0.46

Comments

- The performance results prove that the proposed e-Health system functions in a more efficient manner than existing similar approaches.
- The presented e-Health system is unique in the context of other similar systems considering its distributed architecture, and also the ability to assure complete safety for the collected data, both during the data transmission and processing stages.
- The optimization of the computing core through the implementation of the NTRU-based routines further enhances the system's runtime efficiency, as it is demonstrated by the obtained values of the processing speed.

Comments (2)

- The values of the $XFER_{IN}$ and $XFER_{OUT}$ performance metrics demonstrate the suitability of the system's deployment in the cloud environment, which the data storage and processing backend uses.
- The number of the ciphertexts (N_{CT}) is maintained at the minimum possible level, while the value of the level L is also computed in an optimal fashion.
- The finer the time period granularity is, the greater the amount of the uploaded data becomes.
- The value of level L increases according to an arithmetic model, and it is perfectly balanced relative to the quantity of the encrypted personal health information, which the backend provides as response to the client software module's requests.
- The virtualized 5G data channel is used in an efficient and scalable manner, as demonstrared by the values of the L_{5G} performance metric.

Conclusions

- Flexible and decoupled architecture – the system is capable of accommodating most of the existing and, with a high probability, future client-side data collection devices.
- The e-Health system demonstrates that it is perfectly possible to sustain a completely secure, privacy preserving and resource efficient data management over large amounts of data through the utilization of virtualized 5G data links.
- This case study demonstrates that fully homomorphic encryption is useable in order to secure a software construct like the e-Health system.
- This model can be adapted to any other use case, which involves the processing of large amounts of sensitive data.

Conclusions (2)

- The system is capable to sustain a perfectly functional and secure data flow between the client data collection devices and the data storage and processing backend, in both directions, for the entire sample of 750 citizens from Brasov.
- This contribution proves that the fully homomorphic encryption can be used in order to secure a complex system, while the NTRU-based fully homomorphic encryption model proves that it is useful in order to further enhance the system’s efficient behaviour.

Conclusions (3)

- The system is scalable, which would potentially justify its deployment on full large urban areas.
- The software system's architecture allows for the e-Health platform to be easily extended with new functional capabilities.
- The implementation of the NTRU-based fully homomorphic encryption routines determines an efficient computational behaviour of the system, including the computational time that is used.

Thank You!
Questions and Discussion