Approximate Homomorphic Encryption

MOTIVATION, CONSTRUCTION, APPLICATIONS

YONGSOO SONG

UCSD

What is the Next Goal?

“HE system can evaluate an arbitrary circuit in a polynomial time.”

What is the Next Goal?

"HE system can evaluate an arbitrary circuit in a polynomial time."

Cryptography community has improved the Efficiency of HE system.

- Performance: Speed, Storage, Expansion rate, etc.
- Functionality: Key-switching, Rotation, Plaintext Space, etc.

What is the Next Goal?

"HE system can evaluate an arbitrary circuit in a polynomial time."

Cryptography community has improved the Efficiency of HE system.

- Performance: Speed, Storage, Expansion rate, etc.
- Functionality: Key-switching, Rotation, Plaintext Space, etc.

Reduction of "Gap" between Real and Encrypted computations.

Datatypes and operations

- Boolean Circuit (Bit Operation)
- Integer Operations
- Modular Arithmetic
- Approximate Arithmetic (Fixed/Floating-point Operation)
- Logical Operations (If & Else statement)

Homomorphic Encryption Schemes

Scheme	Plaintext Slot	Good	Bad	Library
"Word Encryption" Brakerski-Gentry-Vaikuntanathan'12 Gentry-Halevi-Smart'12a,b,c Brakerski'12, Fan-Vercauteren'12 Halevi-Shoup'13,14,15	$GF(p^d)$ (Z_p)	Polylog overhead (Amortized time & Expansion rate)	Bootstrapping	HElib SEAL …
Gentry-Sahai-Waters'13				
"Bitwise Encryption" Ducas-Micciancio'15 Chillotti-Gama-Georgieva-Izabachene'16,17				

Homomorphic Encryption Schemes

Scheme	Plaintext Slot	Good	Bad	Library
"Word Encryption" Brakerski-Gentry-Vaikuntanathan'12 Gentry-Halevi-Smart'12a,b,c Brakerski'12, Fan-Vercauteren'12 Halevi-Shoup'13,14,15	$GF(p^d)$ (Z_p)	Polylog overhead (Amortized time & Expansion rate)	Bootstrapping	HElib SEAL …
Gentry-Sahai-Waters'13	Z, Z[X] ({0,1})	Beauty ☺	Inefficient	
"Bitwise Encryption" Ducas-Micciancio'15 Chillotti-Gama-Georgieva-Izabachene'16,17				

Homomorphic Encryption Schemes

Scheme	Plaintext Slot	Good	Bad	Library
"Word Encryption" Brakerski-Gentry-Vaikuntanathan'12 Gentry-Halevi-Smart'12a,b,c Brakerski'12, Fan-Vercauteren'12 Halevi-Shoup'13,14,15	$GF(p^d)$ (Z_p)	Polylog overhead (Amortized time & Expansion rate)	Bootstrapping	HElib SEAL …
Gentry-Sahai-Waters'13	Z, Z[X] ({0,1})	~~Beauty ☺~~ Toolkit for FHEW	Inefficient	
"Bitwise Encryption" Ducas-Micciancio'15 Chillotti-Gama-Georgieva-Izabachene'16,17	{0,1}, ($\{0,1\}^*$)	Evaluation with Bootstrapping. Latency.	Amortized time & Expansion rate	FHEW TFHE

Application Researches of HE (2017~2018)

“Homomorphic Encryption” in ePrint and IEEE Xplore

Application Researches of HE (2017~2018)

“Homomorphic Encryption” in ePrint and IEEE Xplore

◦ Machine Learning:	11	(2018/233,202,139,074, 2017/979,715. SSCI, IEEE Access, IEEE Journal, ICCV, SMARTCOMP)
◦ Neural Network:	2	(2018/073, 2017/1114)
◦ Genomic Data:	7	(2017/955,770,294,228. EUSIPCO, SMARTCOMP, IEEE Journal)
◦ Health Data:	2	(IBM Journal, IEEE Journal)
◦ Biometric Data:	2	(IEEE Access, IEEE Conference)
◦ Energy Management:	3	(2017/1212. IEEE Big Data, IET Journal)
◦ Big Data:	1	(ICBDA)
◦ Advertising:	1	(WIFS)
◦ Internet of Things:	1	(IWCMC)
◦ Election:	1	(2017/166)

Application Researches of HE (2017~2018)

"Homomorphic Encryption" in ePrint and IEEE Xplore

Machine Learning:	11	(2018/233,202,139,074, 2017/979,715. SSCI, IEEE Access, IEEE Journal, ICCV, SMARTCOMP)
Neural Network:	2	(2018/073, 2017/1114)
Genomic Data:	7	(2017/955,770,294,228. EUSIPCO, SMARTCOMP, IEEE Journal)
Health Data:	2	(IBM Journal, IEEE Journal)
Biometric Data:	2	(IEEE Access, IEEE Conference)
Energy Management:	3	(2017/1212. IEEE Big Data, IET Journal)
Big Data:	1	(ICBDA)
Advertising:	1	(WIFS)
Internet of Things:	1	(IWCMC)
Election:	1	(2017/166)

How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?

How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?

Word Encryption

- Represent a real number as an integer.
- No Rounding operation is very expensive.

How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?

Word Encryption

- Represent a real number as an integer.
- No Rounding operation is very expensive.
- Bit size of message grows exponentially.

1,234 * 689 * 2,194 * 917 * 3,323 * 4,154 * 489 * 3,772 = 4,355,296,408,921,213,975,719,328 > 2^{85}.

How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?

Word Encryption

- Represent a real number as an integer.
- No Rounding operation is very expensive.
- Bit size of message grows exponentially.

1,234 * 689 * 2,194 * 917 * 3,323 * 4,154 * 489 * 3,772 = 4,355,296,408,921,213,975,719,328 > 2^{85}.

Base Encoding [DGL+'15, CSCW'16, CLPX'17 (High-precision HE)]

- Express a real number as a (small) polynomial. e.g. (1.234) → ($1 + 2X^{-1} + 3X^{-2} + 4X^{-3}$)

How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?

Word Encryption

- Represent a real number as an integer.
- No Rounding operation is very expensive.
- Bit size of message grows exponentially.

1,234 * 689 * 2,194 * 917 * 3,323 * 4,154 * 489 * 3,772 = 4,355,296,408,921,213,975,719,328 > 2^{85}.

Base Encoding [DGL+'15, CSCW'16, CLPX'17 (High-precision HE)]

- Express a real number as a (small) polynomial. e.g. (1.234) → $(1 + 2X^{-1} + 3X^{-2} + 4X^{-3})$
- Exponential growth of Degree
- Trade-off between Precision & Number of slots
- No Bootstrapping

How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?

Bitwise Encryption

- 0.06 sec for (2-to-1) gate.
- 10 sec for (6-to-6) circuit.

How to perform Approximate Arithmetic on HE?

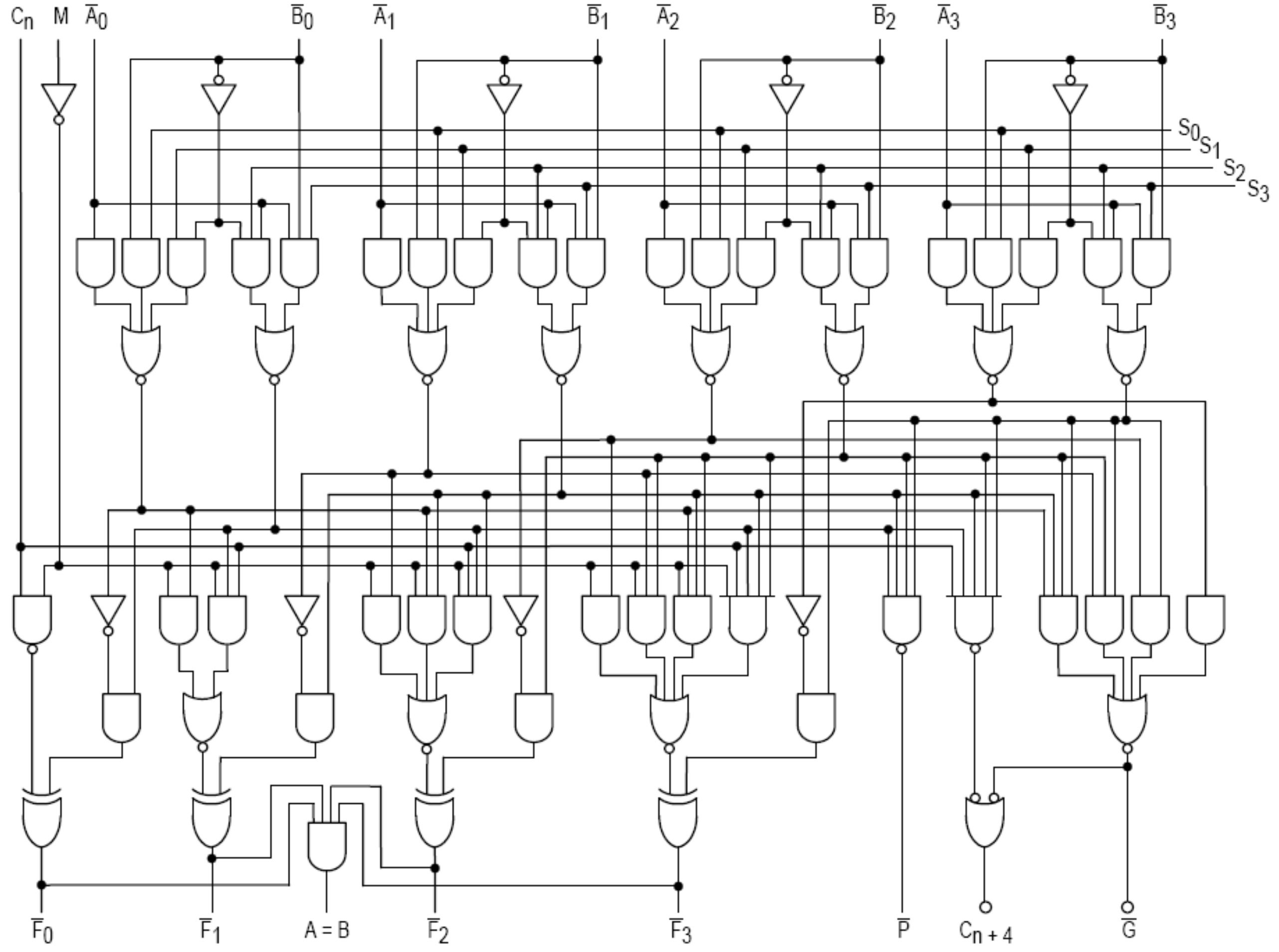
1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?

Bitwise Encryption

- 0.06 sec for (2-to-1) gate.
- 10 sec for (6-to-6) circuit.

75 Gates for an operation on two four-bit strings.

How many gates for
16-bit / 32-bit precision multiplication?



Homomorphic Encryption Schemes

Scheme	Plaintext Slot	Good	Bad	Library
"Word Encryption" Brakerski-Gentry-Vaikuntanathan'12 Gentry-Halevi-Smart'12a,b,c Brakerski'12, Fan-Vercauteren'12 Halevi-Shoup'13,14,15	$GF(p^d)$ (Z_p)	Polylog overhead (Amortized time & Expansion rate)	Bootstrapping	HElib SEAL …
Gentry-Sahai-Waters'13	Z, Z[X] ({0,1})	~~Beauty~~ 😃 Toolkit for FHEW	Inefficient	
"Bitwise Encryption" Ducas-Micciancio'15 Chillotti-Gama-Georgieva-Izabachene'16,17	{0,1}, ($\{0,1\}^k$)	Evaluation with Bootstrapping. Latency.	Amortized time & Expansion rate	FHEW TFHE

Homomorphic Encryption Schemes

Scheme	Plaintext Slot	Good	Bad	Library
"Word Encryption" Brakerski-Gentry-Vaikuntanathan'12 Gentry-Halevi-Smart'12a,b,c Brakerski'12, Fan-Vercauteren'12 Halevi-Shoup'13,14,15	$GF(p^d)$ (Z_p)	Polylog overhead (Amortized time & Expansion rate)	Bootstrapping	HElib SEAL …
Gentry-Sahai-Waters'13	Z, Z[X] ({0,1})	~~Beauty ☺~~ Toolkit for FHEW	Inefficient	
"Bitwise Encryption" Ducas-Micciancio'15 Chillotti-Gama-Georgieva-Izabachene'16,17	{0,1}, ($\{0,1\}^k$)	Evaluation with Bootstrapping. Latency.	Amortized time & Expansion rate	FHEW TFHE
"Approximate Encryption" Cheon-Kim-Kim-Song'17 Cheon-Han-Kim-Kim-Song'18	Complex (Real) Numbers	Fixed-point Arithmetic. Polylog overhead.		HEAAN (慧 眼)

Approximate Homomorphic Encryption

Motivation: Imitate the approximate arithmetic on computer system.

$$1.234 * 0.689 = (1{,}234 * 10^{-3}) * (689 * 10^{-3})$$

Approximate Homomorphic Encryption

Motivation: Imitate the approximate arithmetic on computer system.

$$1.234 * 0.689 = (1{,}234 * 10^{-3}) * (689 * 10^{-3}) = 850{,}226 * 10^{-6} = 850 * 10^{-3}$$

Approximate Homomorphic Encryption

Motivation: Imitate the approximate arithmetic on computer system.

$$1.234 * 0.689 = (1{,}234 * 10^{-3}) * (689 * 10^{-3}) = 850{,}226 * 10^{-6} = 850 * 10^{-3}$$

Idea 1. Every number contains an Approximation Error (between unknown true value).

Consider an RLWE error as part of it.

Approximate Homomorphic Encryption

Motivation: Imitate the approximate arithmetic on computer system.

$$1.234 * 0.689 = (1{,}234 * 10^{-3}) * (689 * 10^{-3}) = 850{,}226 * 10^{-6} = 850 * 10^{-3}$$

Idea 1. Every number contains an Approximation Error (between unknown true value).

Consider an RLWE error as part of it.

$$ct = \mathrm{Enc}\,(m) \quad \text{if} \quad [\langle ct, sk \rangle]_q = m+e \approx m.$$

Approximate Homomorphic Encryption

Motivation: Imitate the approximate arithmetic on computer system.

$1.234 * 0.689 = (1{,}234 * 10^{-3}) * (689 * 10^{-3}) = 850{,}226 * 10^{-6} = 850 * 10^{-3}$

Idea 1. Every number contains an Approximation Error (between unknown true value).

Consider an RLWE error as part of it.

ct = Enc (m) if $[\langle ct, sk\rangle]_q = m + e \approx m$.

Approximate HE: (1.234) ⇒ (scale by $p=10^4$) ⇒ 12,340.

⇒ (Encrypt) ⇒ $[\langle ct, sk\rangle]_q$ = 12,342 ≈ $1.234 * 10^4$.

Approximate Homomorphic Encryption

Motivation: Imitate the approximate arithmetic on computer system.

$1.234 * 0.689 = (1{,}234 * 10^{-3}) * (689 * 10^{-3}) = 850{,}226 * 10^{-6} = 850 * 10^{-3}$

Idea 2. Approximate Rounding is easy!

$\langle ct, sk \rangle = m \pmod q$

$ct \mapsto ct' = \lceil p^{-1} * ct \rfloor$

$\langle ct', sk \rangle \pmod{p^{-1} q} \approx p^{-1} m$

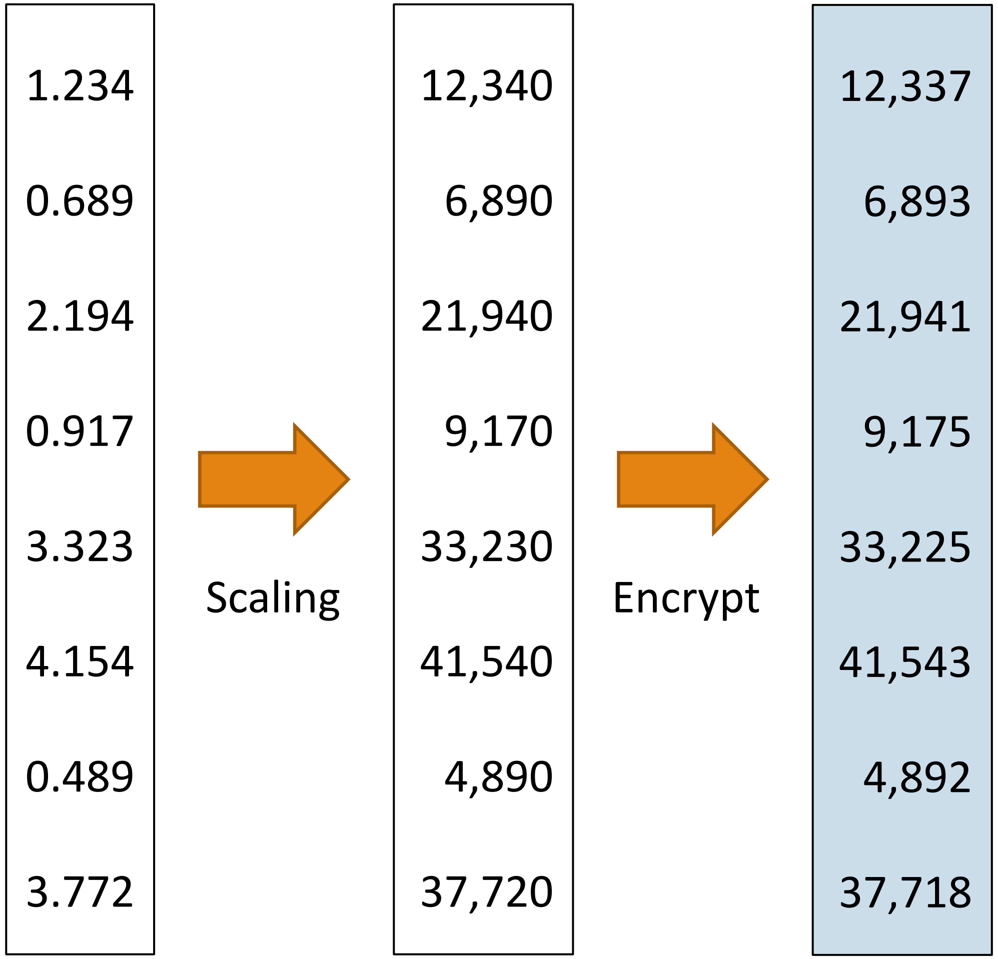
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?

1.234
0.689
2.194
0.917
3.323
4.154
0.489
3.772

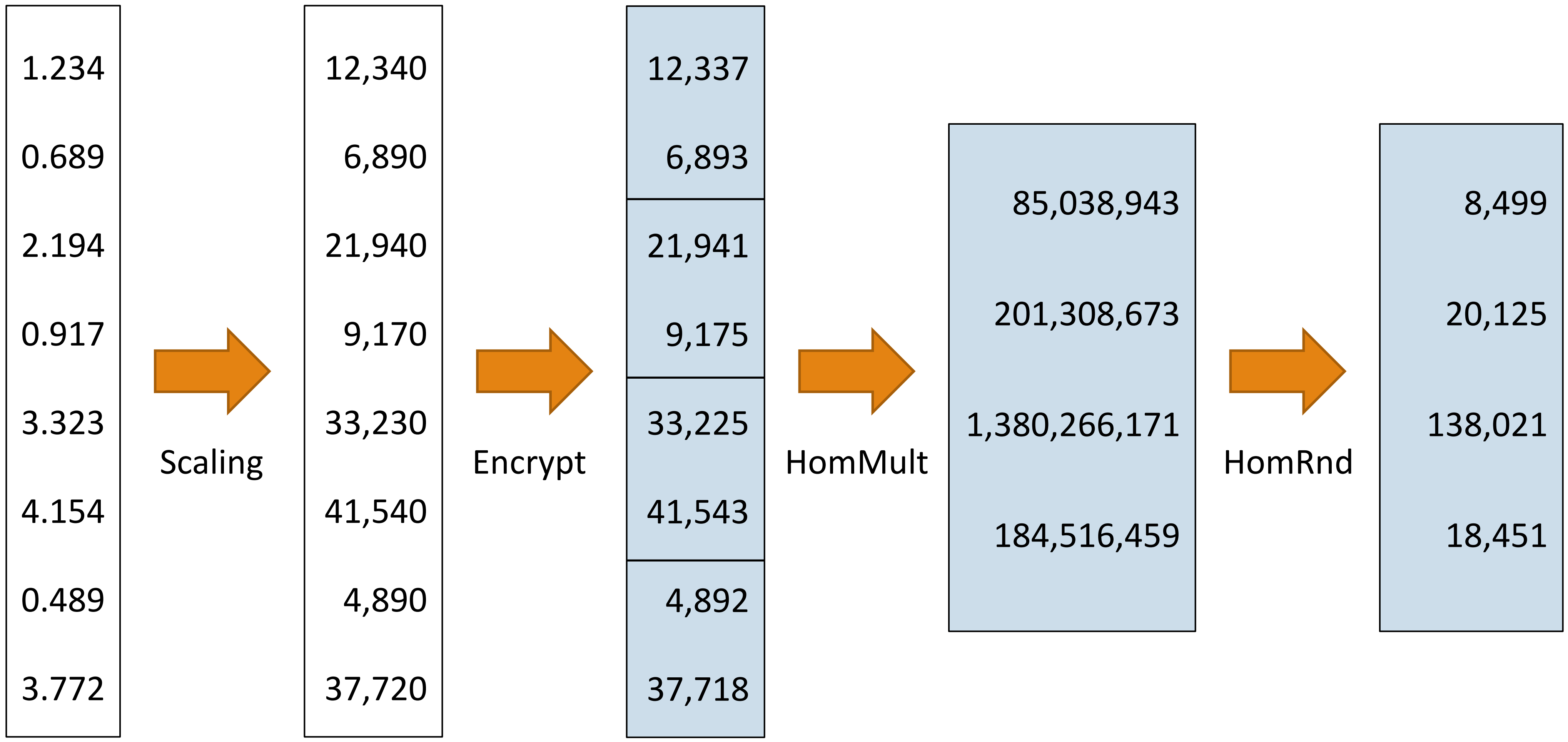
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?



How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?



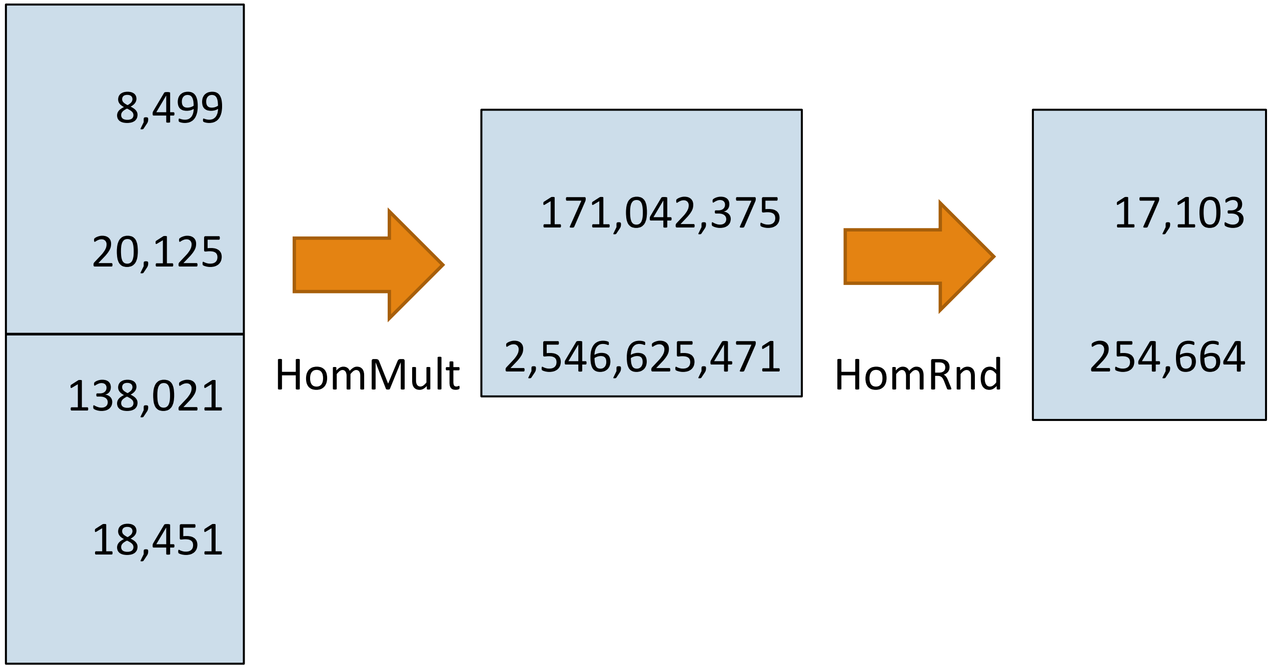
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?

8,499 20,125
138,021 18,451

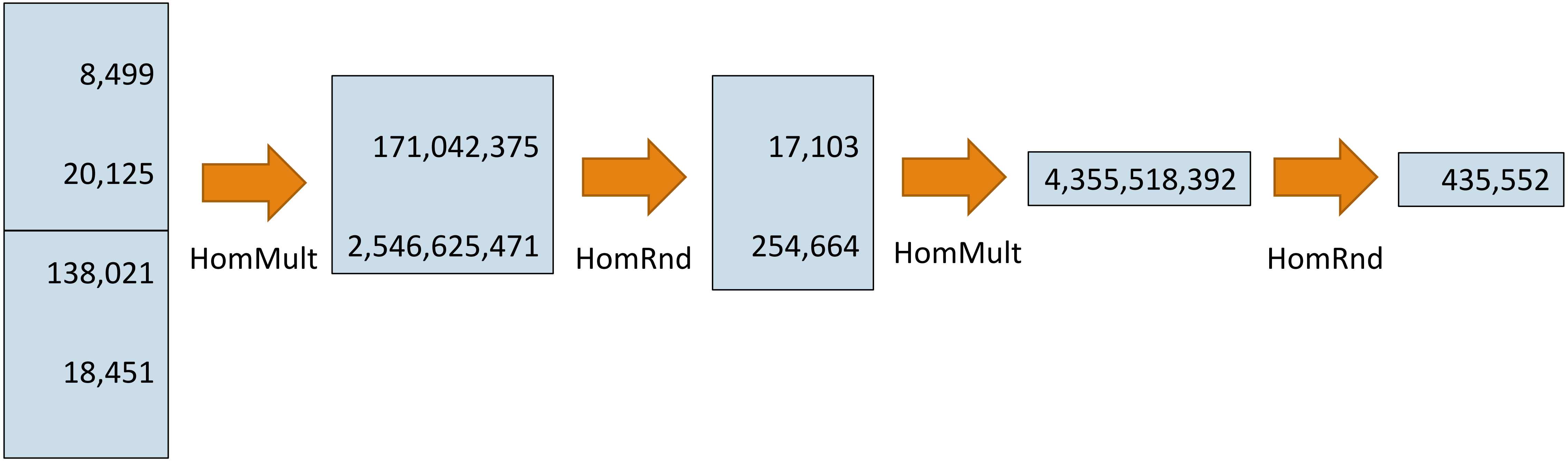
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?



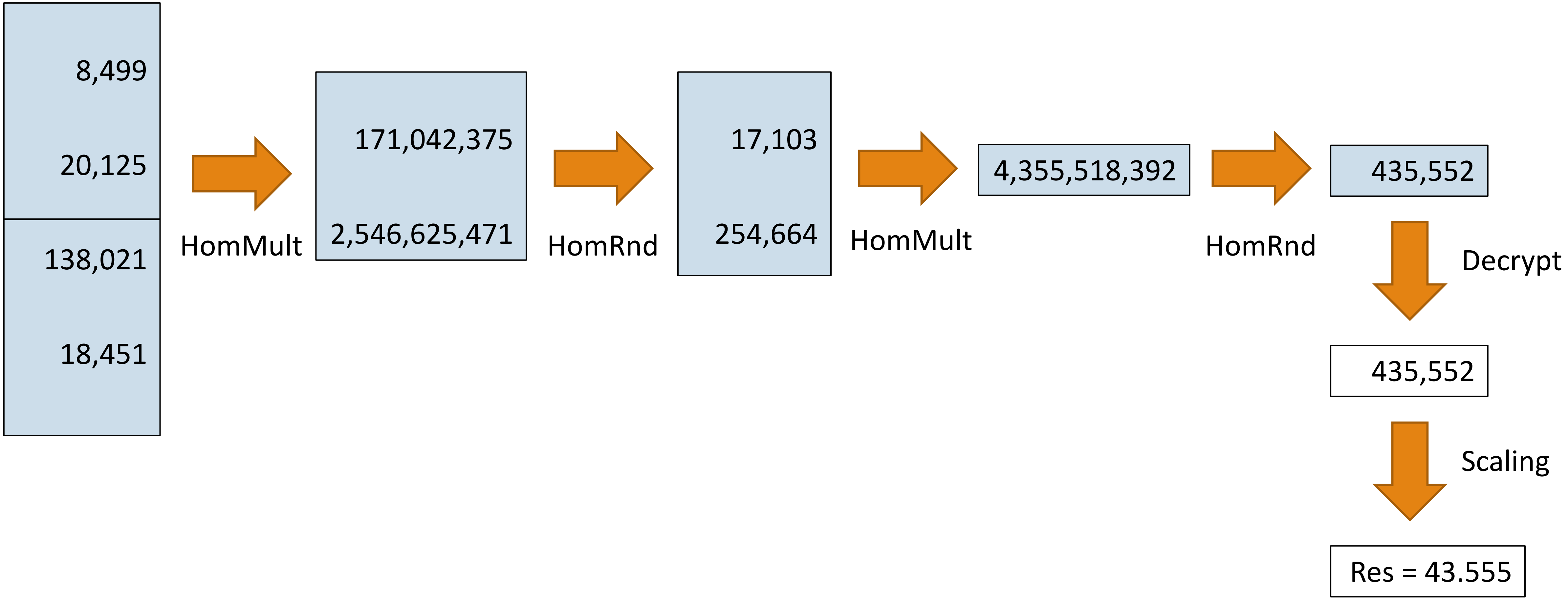
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?



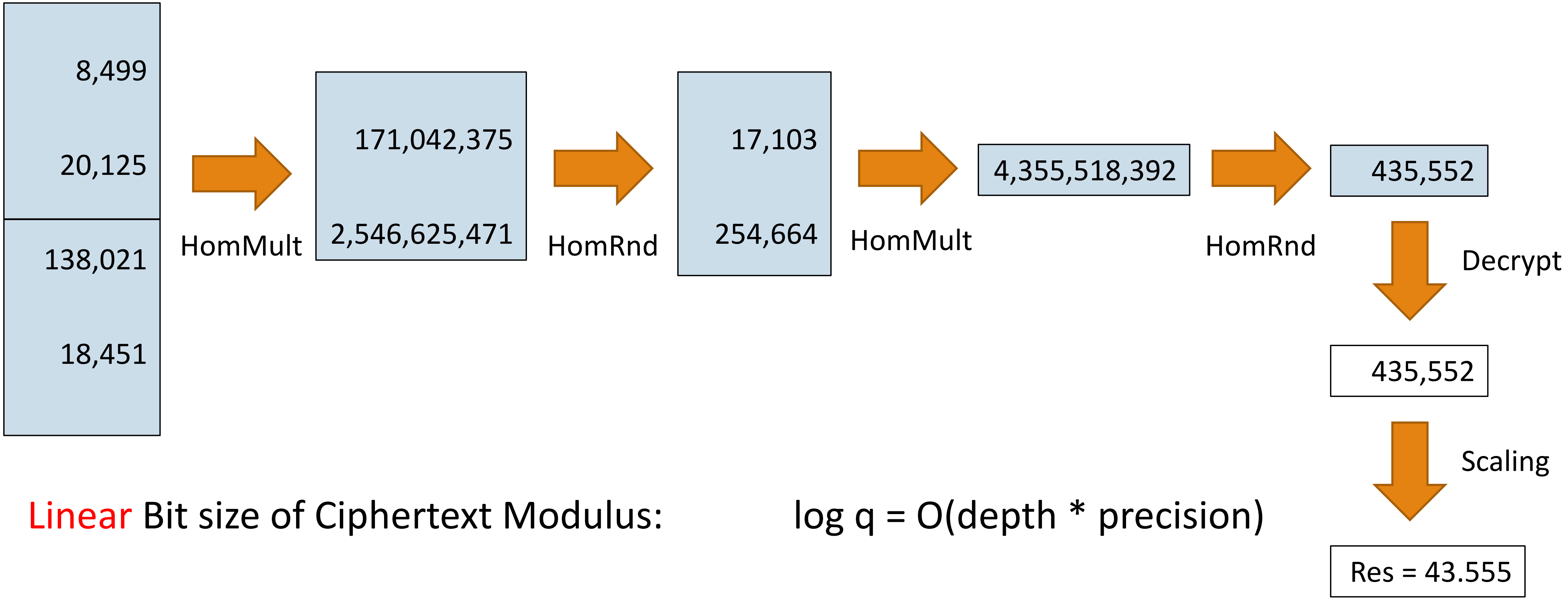
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = 43.555



How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = 43.555



Linear Bit size of Ciphertext Modulus: log q = O(depth * precision)

Functionality of Approximate HE

Packing Technique

- $R = Z[X] / (\Phi_m(X))$.
- $\Phi(X) = \prod_i (X - \zeta_i)$ where ζ_i's are m-th roots of unity.
- Encoding map: from $(M_i)_i$ to $M(X)$ such that $M(\zeta_i) = M_i$

Functionality of Approximate HE

Packing Technique

- $R = Z[X] / (\Phi_m(X))$.
- $\Phi(X) = \prod_i (X - \zeta_i)$ where ζ_i's are m-th roots of unity.
- Encoding map: from $(M_i)_i$ to $M(X)$ such that $M(\zeta_i) = M_i$

Rotation, Conjugation

- Evaluation of $\sigma(X)=X^k$ in Gal($K=Q[X]/(X^N+1)$ / Q).
- Based on the key-switching technique.

Functionality of Approximate HE

Packing Technique

- $R=Z[X] / (\Phi_m(X))$.
- $\Phi(X) = \prod_i (X- \zeta_i)$ where ζ_i's are m-th roots of unity.
- Encoding map: from $(M_i)_i$ to $M(X)$ such that $M(\zeta_i) = M_i$

Rotation, Conjugation

- Evaluation of $\sigma(X)=X^k$ in $Gal(K=Q[X]/(X^N+1) / Q)$.
- Based on the key-switching technique.

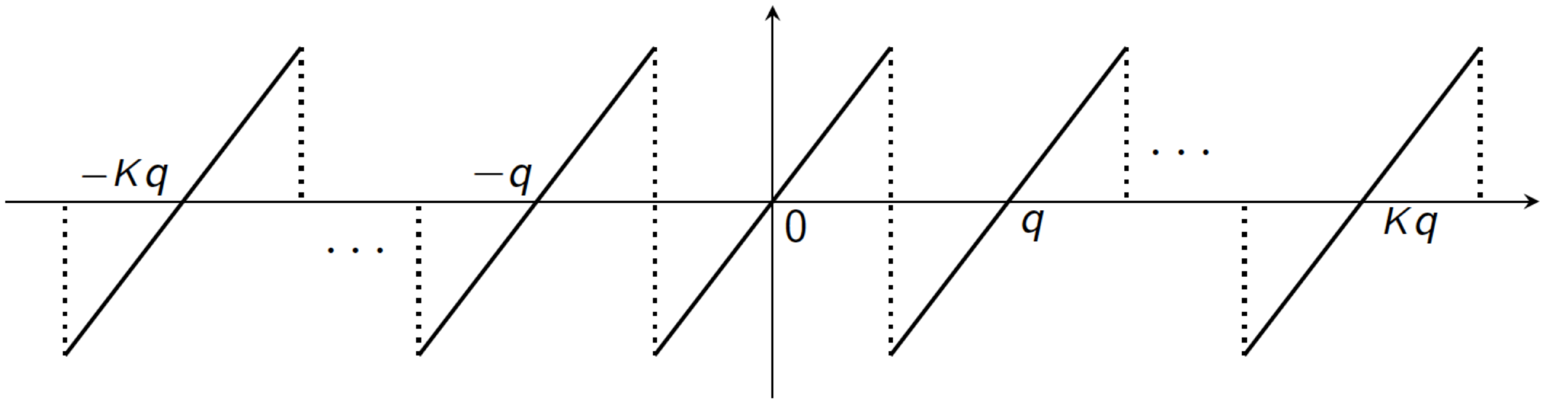
Evaluation of Analytic Functions

- $\exp (z)$,
- z^{-1}

Bootstrapping for the Approximate HE (EC'18)

Decryption circuit

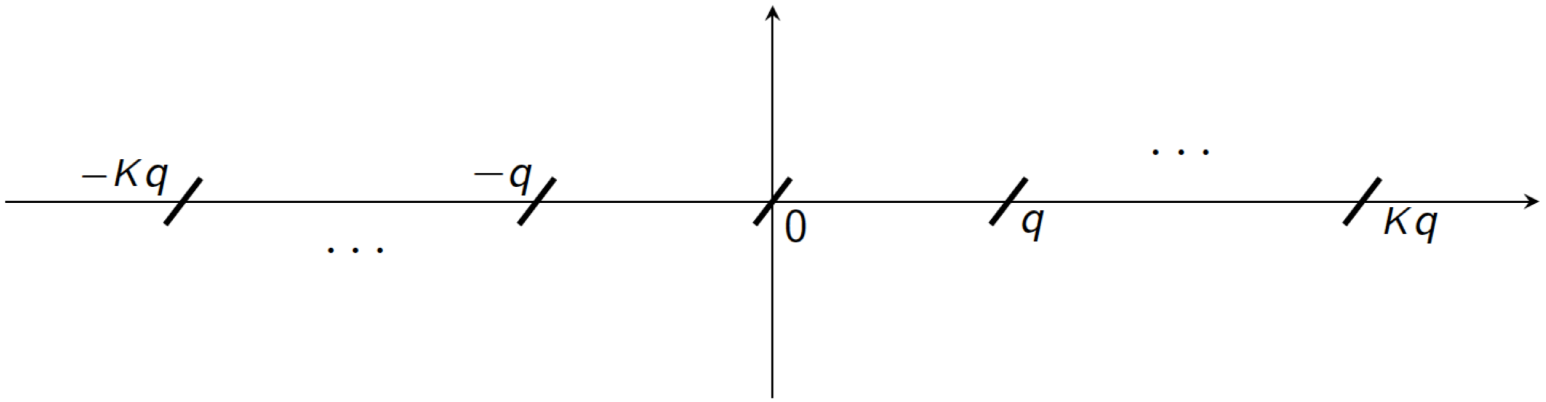
- M = <ct, sk> (mod q).
- Goal: Represent modular reduction as a circuit over the complex numbers.



Bootstrapping for the Approximate HE (EC'18)

Decryption circuit

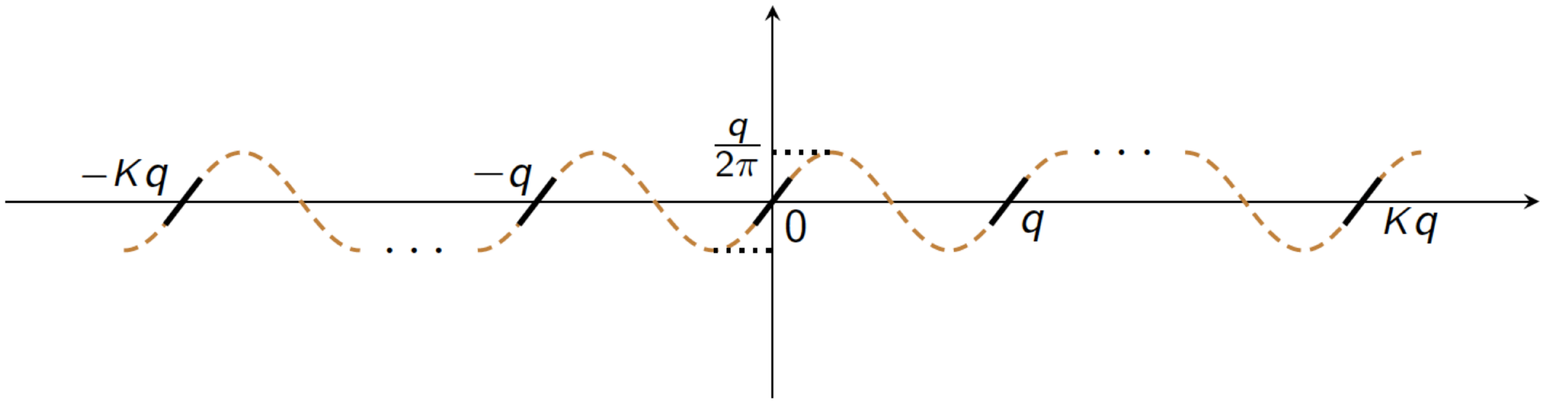
- M = <ct, sk> (mod q).
- Goal: Represent modular reduction as a circuit over the complex numbers.



Bootstrapping for the Approximate HE (EC'18)

Decryption circuit

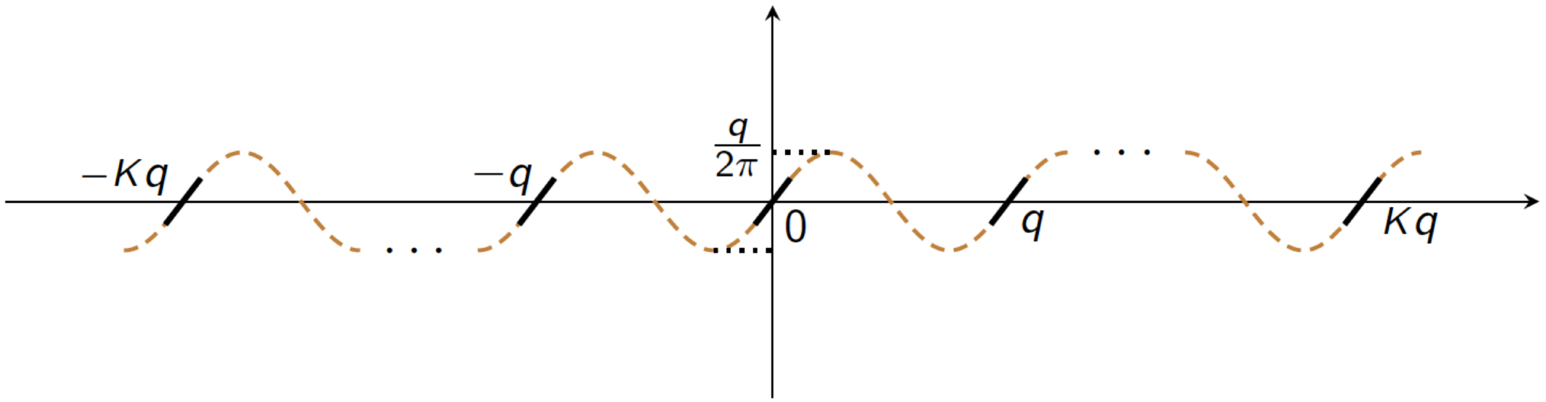
- ~~M = <ct, sk> (mod q).~~ M ≈ (q/2π) sin θ, θ = (2π/q) <ct,sk>.
- Goal: Represent modular reduction as a circuit over the complex numbers.



Bootstrapping for the Approximate HE (EC'18)

Decryption circuit

- ~~$M = \langle ct, sk \rangle \pmod q$.~~ $M \approx (q/2\pi) \sin \theta$, $\theta = (2\pi/q) \langle ct,sk \rangle$.
- Goal: Represent modular reduction as a circuit over the complex numbers.



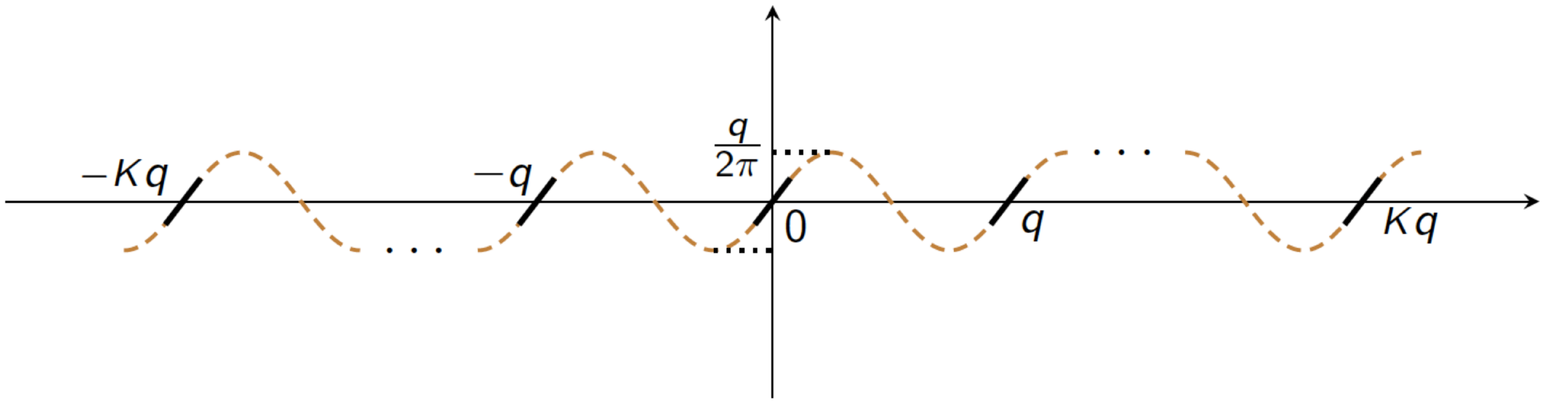
Evaluation of sine

- $\cos \theta = \cos^2(\theta/2) - \sin^2(\theta/2)$, $\sin\theta = 2 \cos(\theta/2) \sin(\theta/2)$.

Bootstrapping for the Approximate HE (EC'18)

Decryption circuit

- ~~M = <ct, sk> (mod q).~~ M ≈ (q/2π) sin θ, θ = (2π/q) <ct,sk>.
- Goal: Represent modular reduction as a circuit over the complex numbers.



Evaluation of sine

- $\cos\theta = \cos^2(\theta/2) - \sin^2(\theta/2)$, $\sin\theta = 2\cos(\theta/2)\sin(\theta/2)$.
- From $[-2K\pi/2^r, 2K\pi/2^r]$ to $[-2K\pi, 2K\pi]$.
- Linear Complexity for Modulus Reduction Operation!
- <ct',sk> (mod Q) ≈ M

Following Work

Privacy-preserving Training of Logistic Regression Model

- Kim-Song-Wang-Xia-Jiang, JMIR Med Inform'18
- Kim-Song-Kim-Lee-Cheon, iDASH P&S Workshop'17, BMC Med Genomics'18 (in submission).

  e.g. Six minutes to obtain a LR model from dataset of size 1579 * (18+1).
- (ongoing) ML based on the financial data with Bootstrapping.

Following Work

Privacy-preserving Training of Logistic Regression Model

- Kim-Song-Wang-Xia-Jiang, JMIR Med Inform'18
- Kim-Song-Kim-Lee-Cheon, iDASH P&S Workshop'17, BMC Med Genomics'18 (in submission).

  e.g. Six minutes to obtain a LR model from dataset of size 1579 * (18+1).
- (ongoing) ML based on the financial data with Bootstrapping.

A Full-RNS Variant of Approx-HE

- Double-CRT (RNS+NTT) representation.
- Implementation without CRT composition or big-integer library.
- Based on the use of approximate basis & approximate modulus switching.

Following Work

Privacy-preserving Training of Logistic Regression Model

- Kim-Song-Wang-Xia-Jiang, JMIR Med Inform'18
- Kim-Song-Kim-Lee-Cheon, iDASH P&S Workshop'17, BMC Med Genomics'18 (in submission).

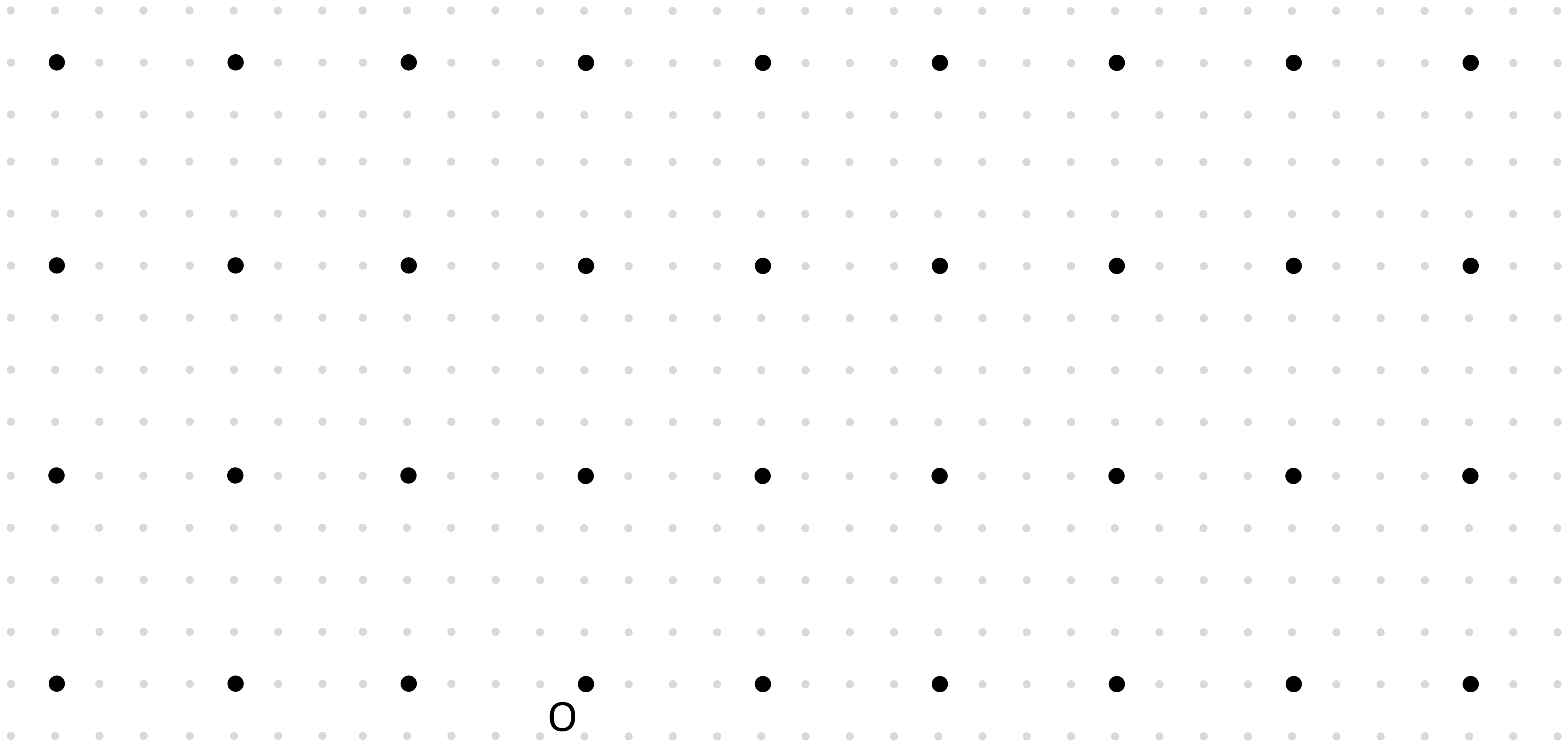
  e.g. Six minutes to obtain a LR model from dataset of size 1579 * (18+1).
- (ongoing) ML based on the financial data with Bootstrapping.

A Full-RNS Variant of Approx-HE

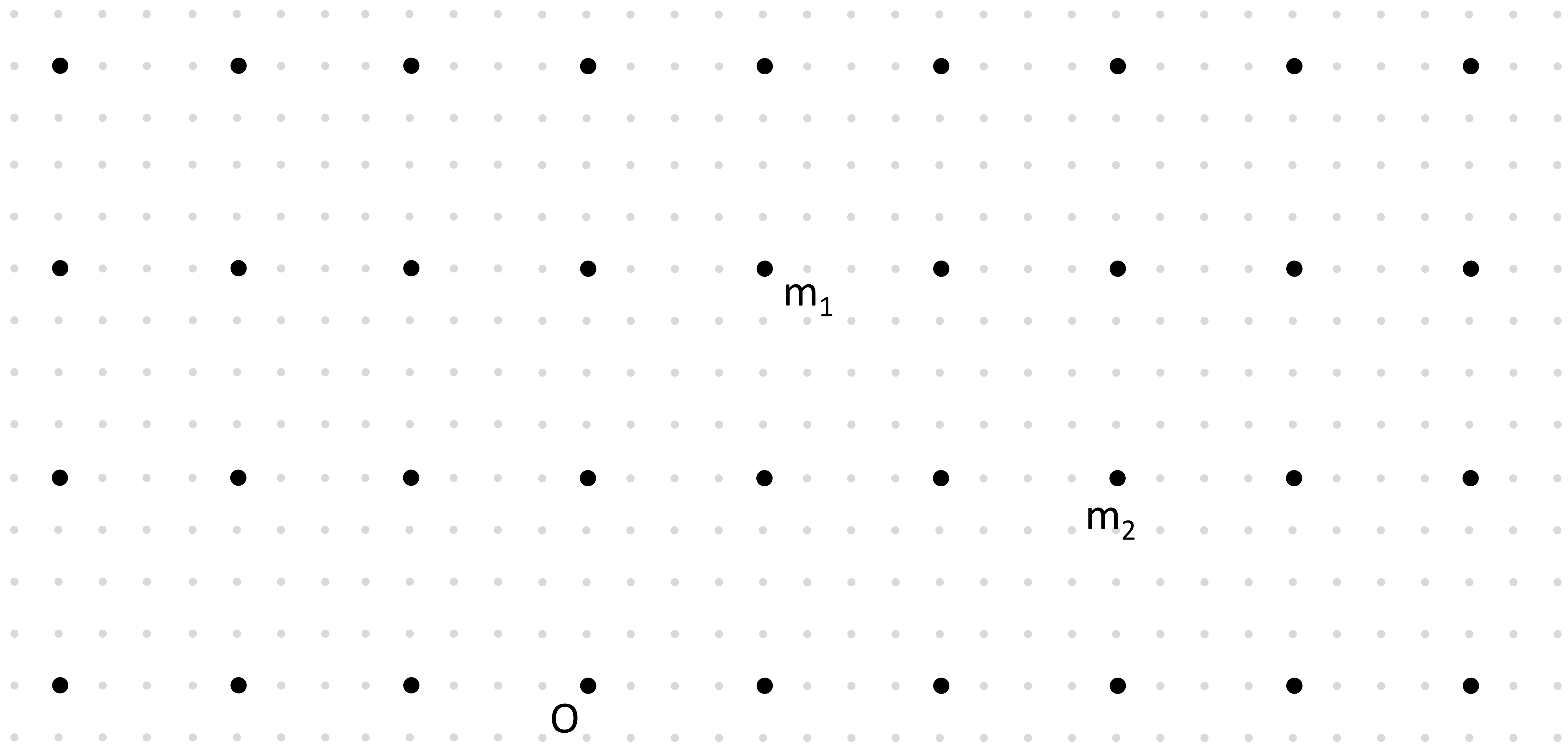
- Double-CRT (RNS+NTT) representation.
- Implementation without CRT composition or big-integer library.
- Based on the use of approximate basis & approximate modulus switching.

Open problems??

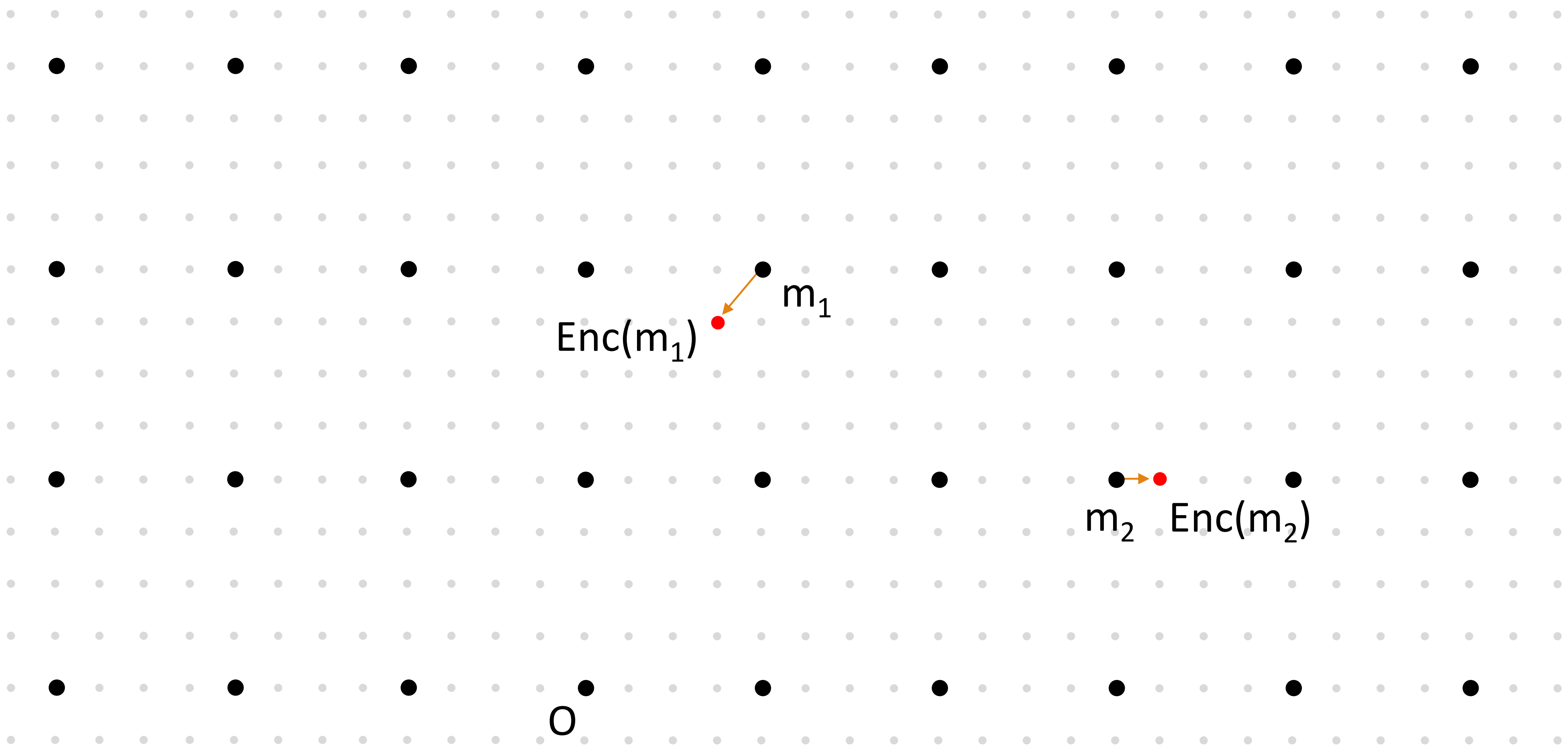
Homomorphic Encryption Framework



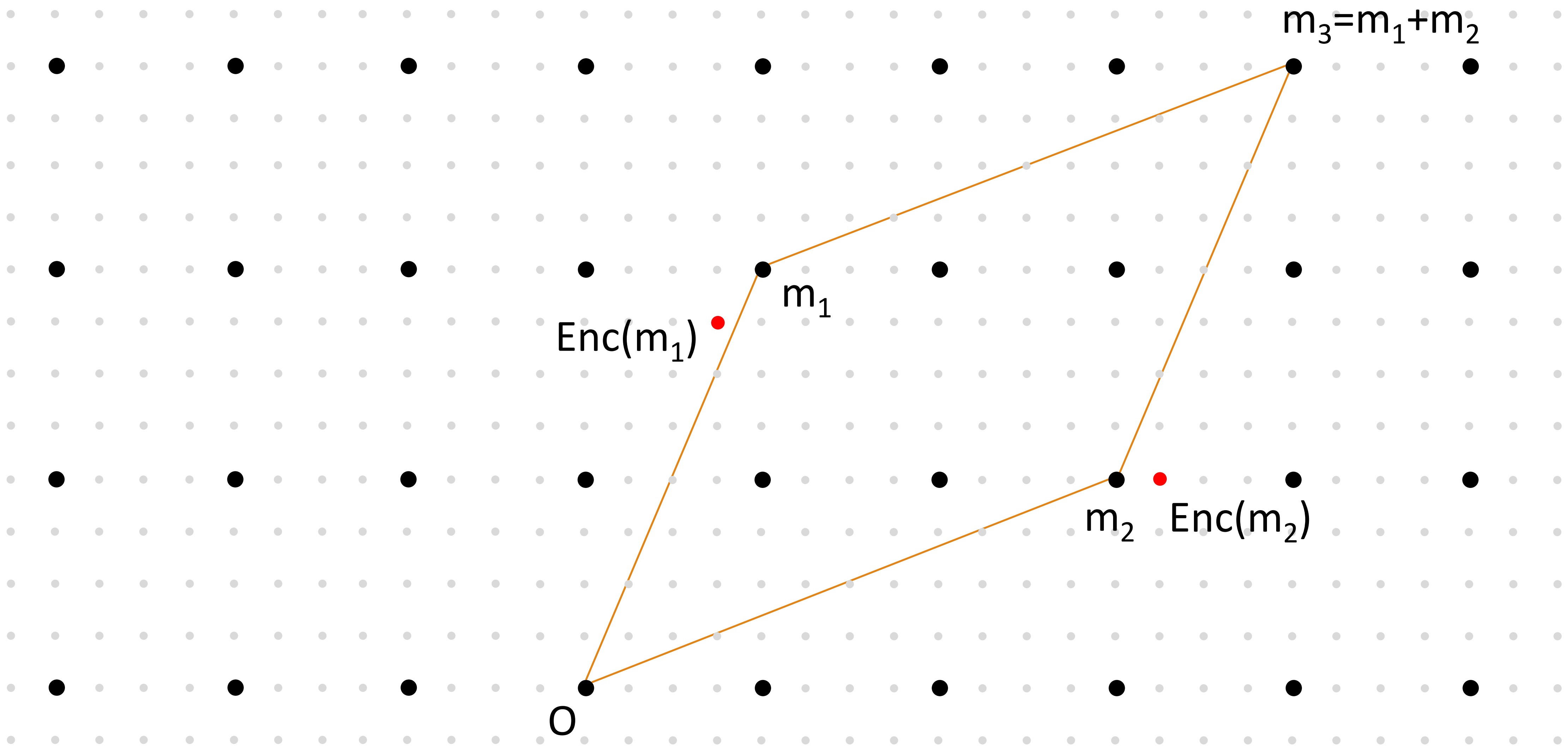
Homomorphic Encryption Framework



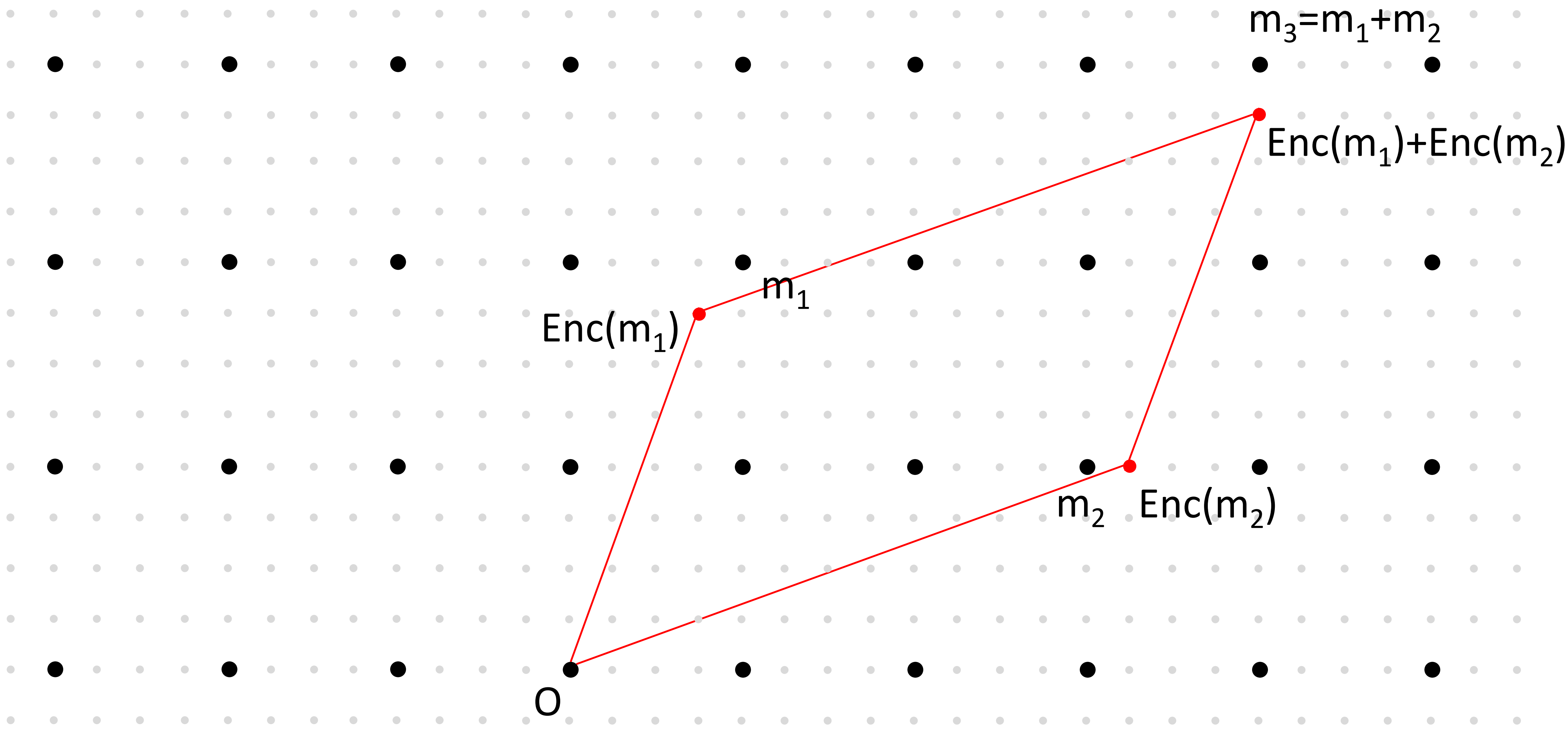
Homomorphic Encryption Framework (Encryption)



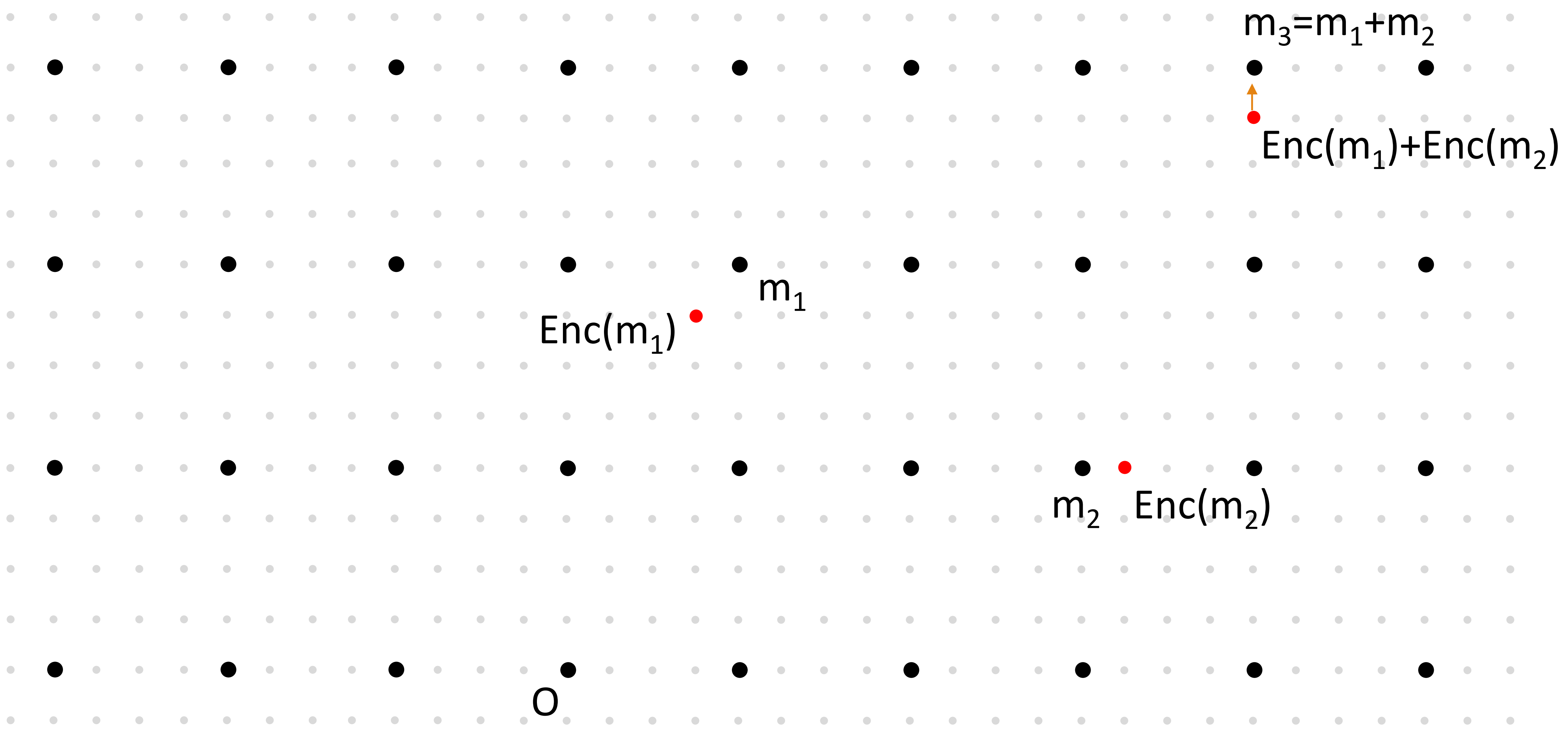
Homomorphic Encryption Framework (Addition)



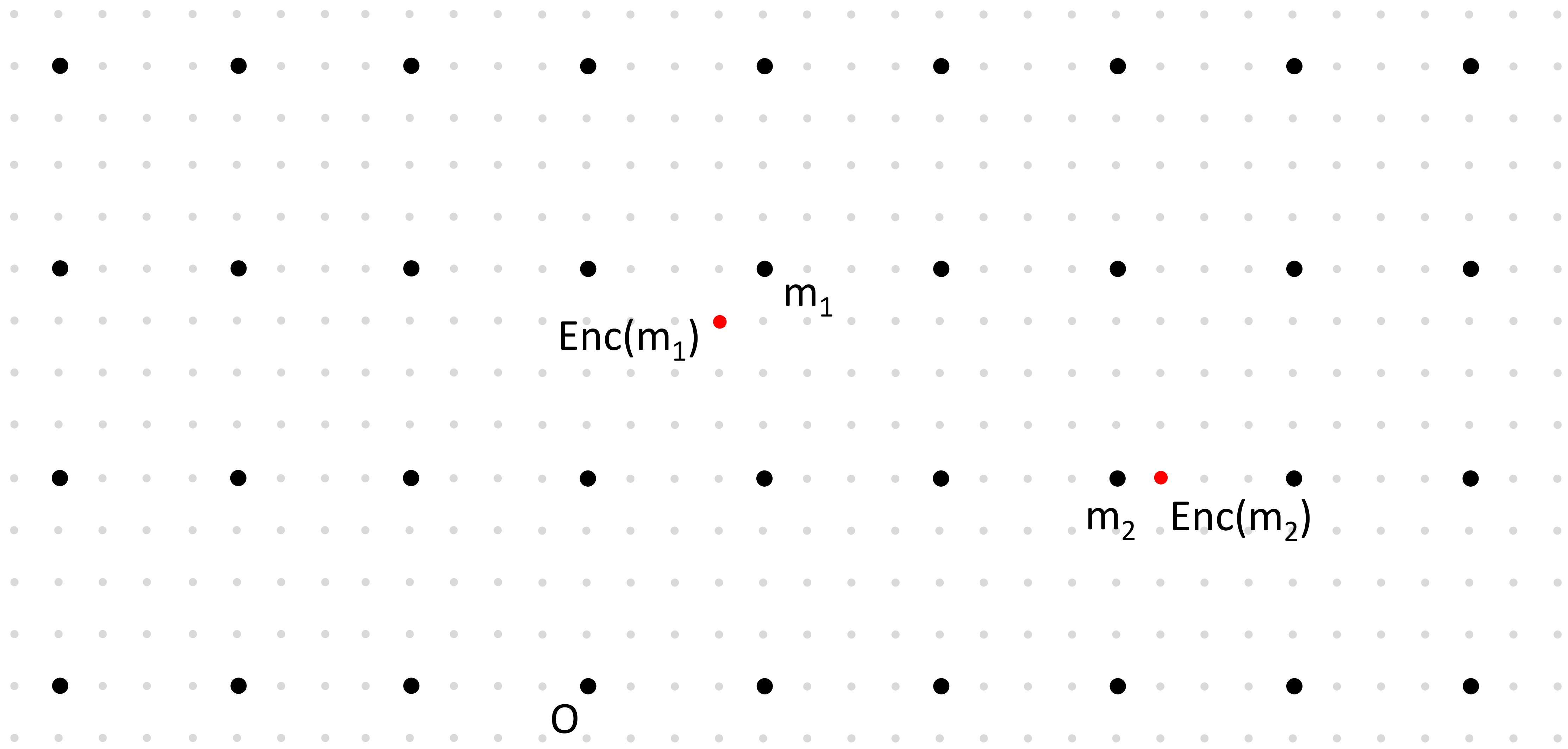
Homomorphic Encryption Framework (Addition)



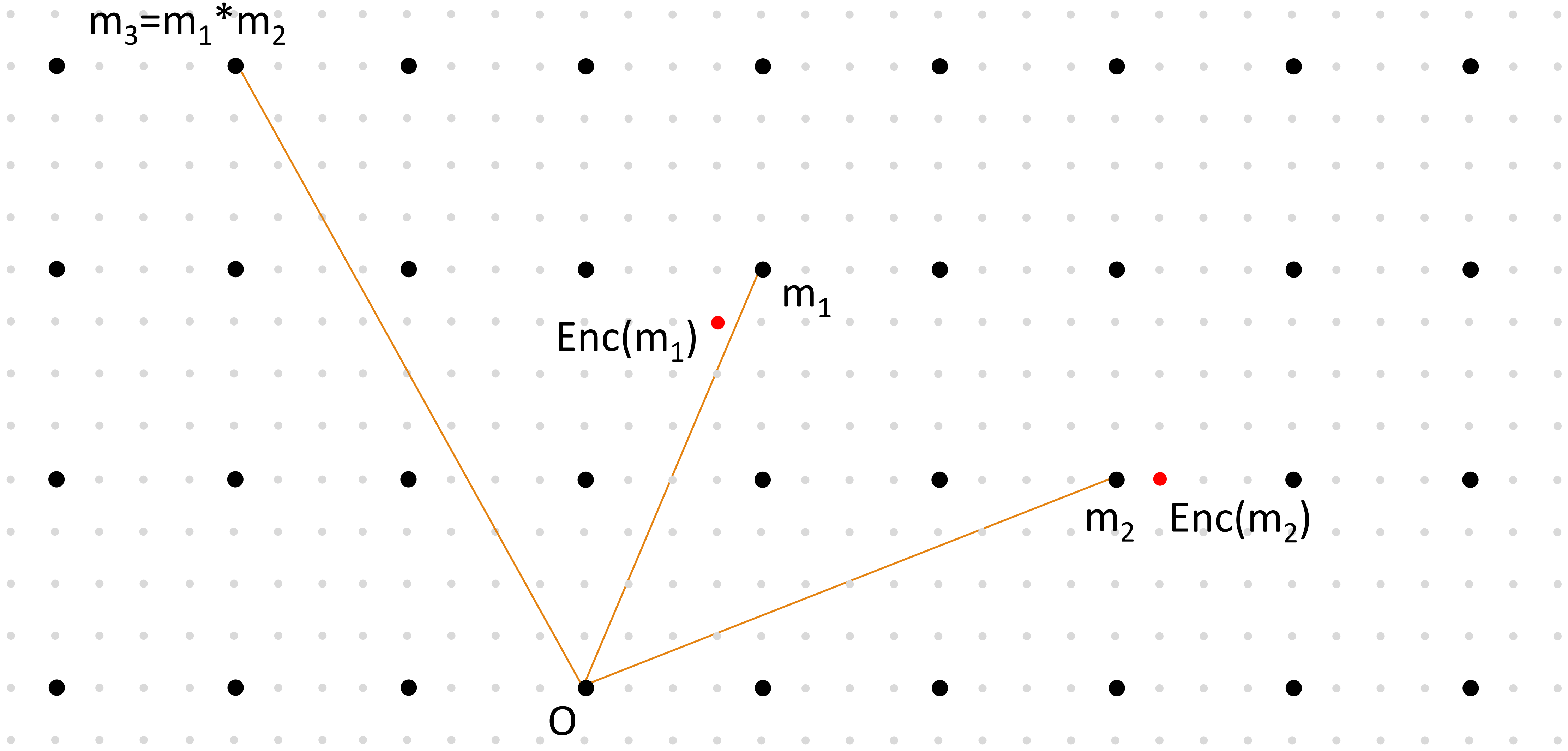
Homomorphic Encryption Framework (Addition)



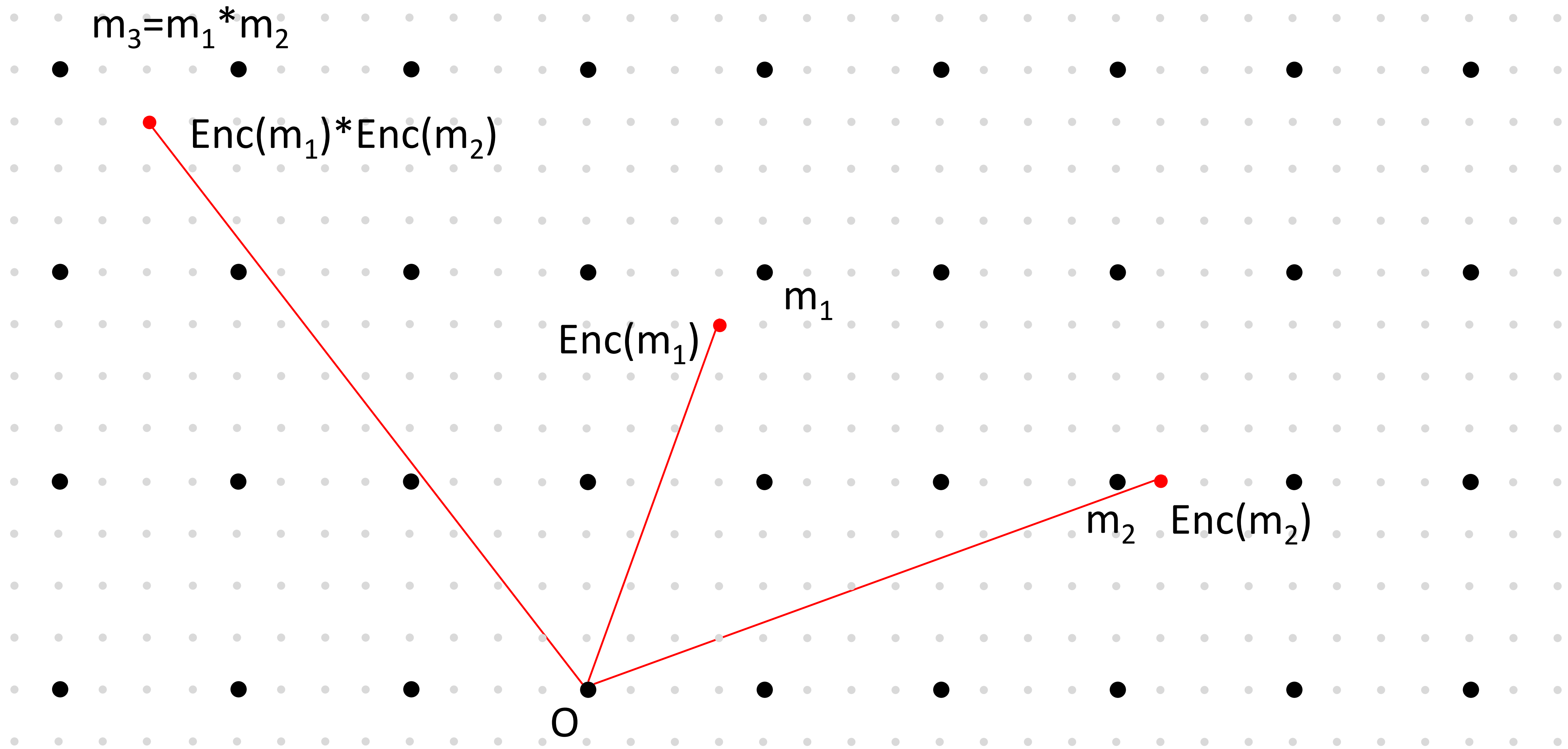
Homomorphic Encryption Framework (Multiplication)



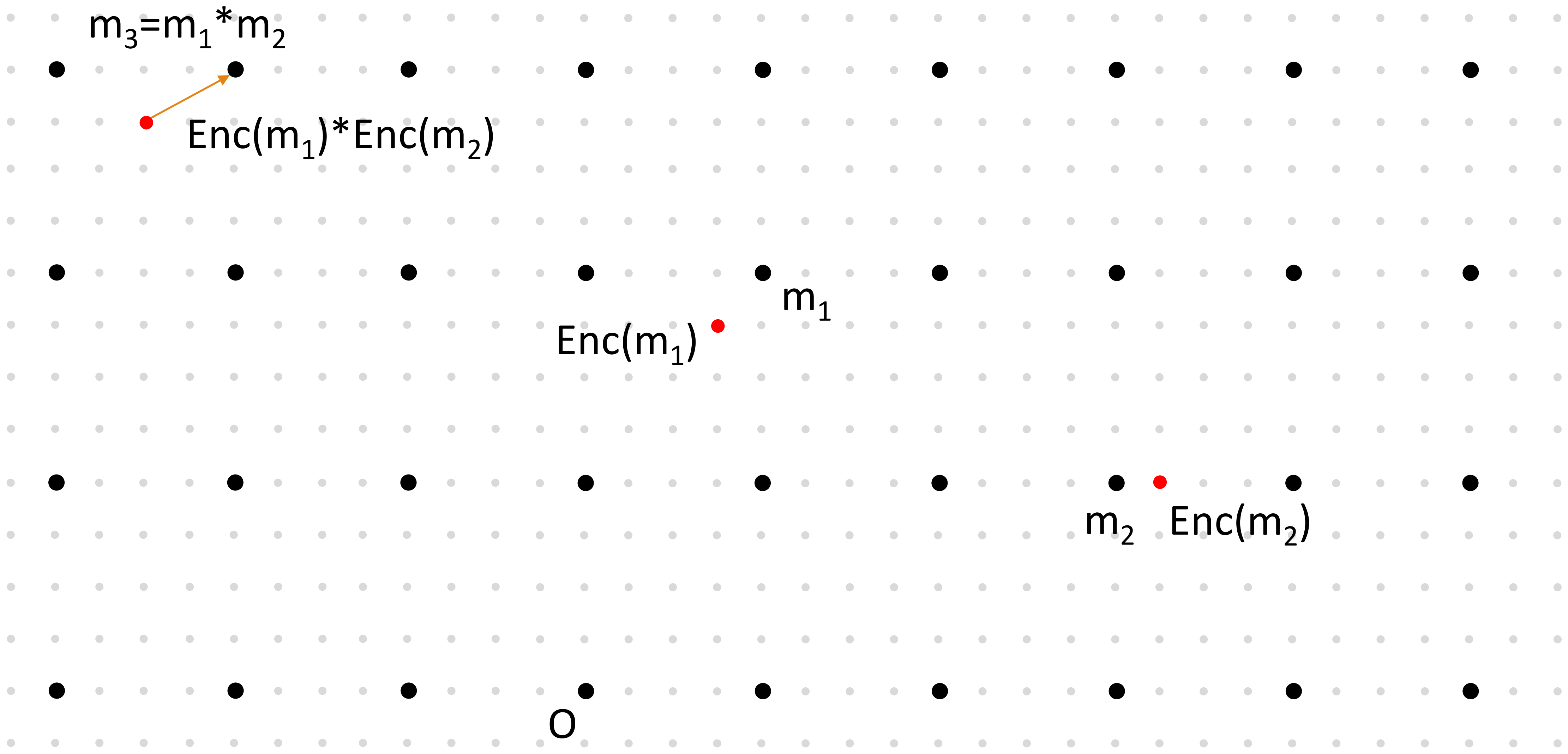
Homomorphic Encryption Framework (Multiplication)



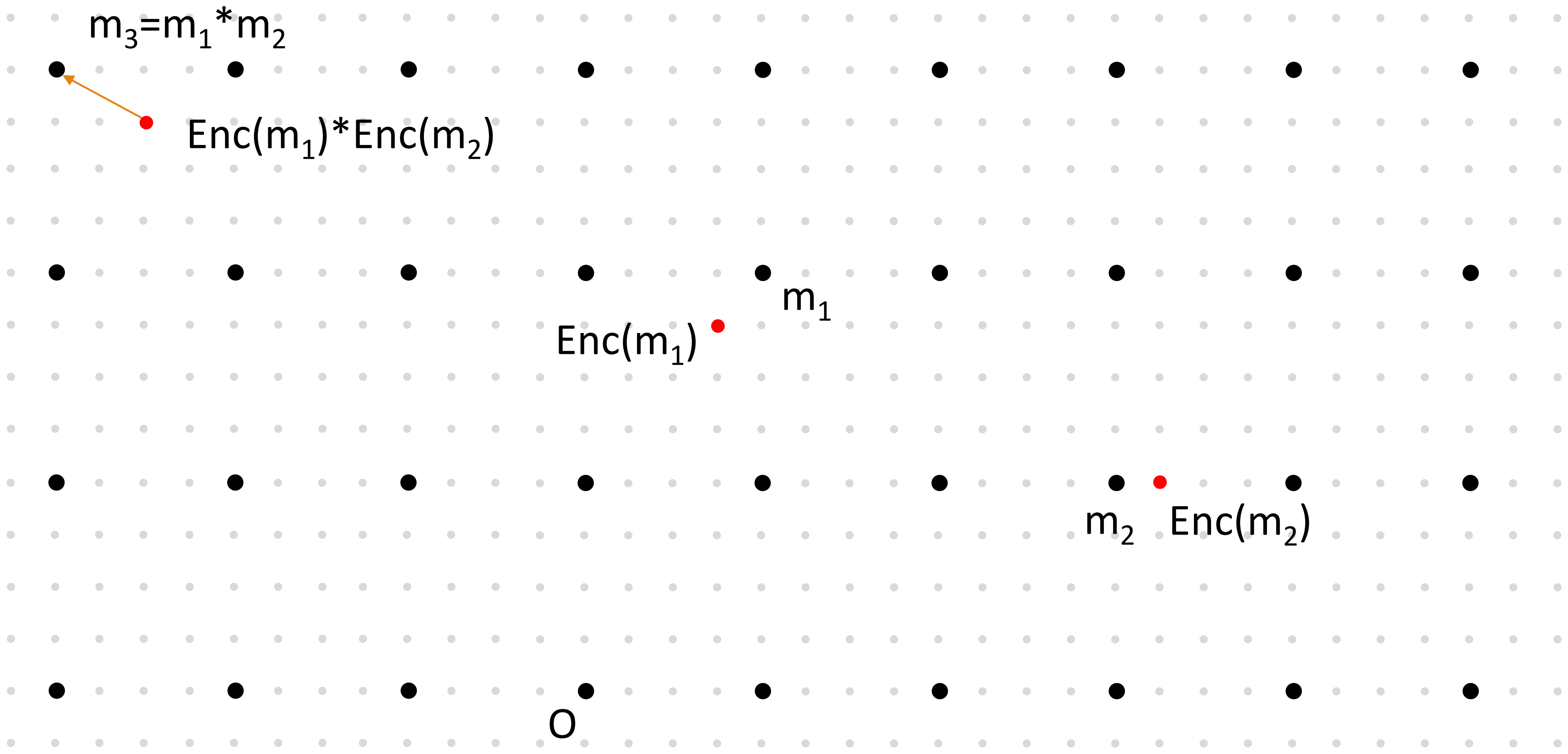
Homomorphic Encryption Framework (Multiplication)



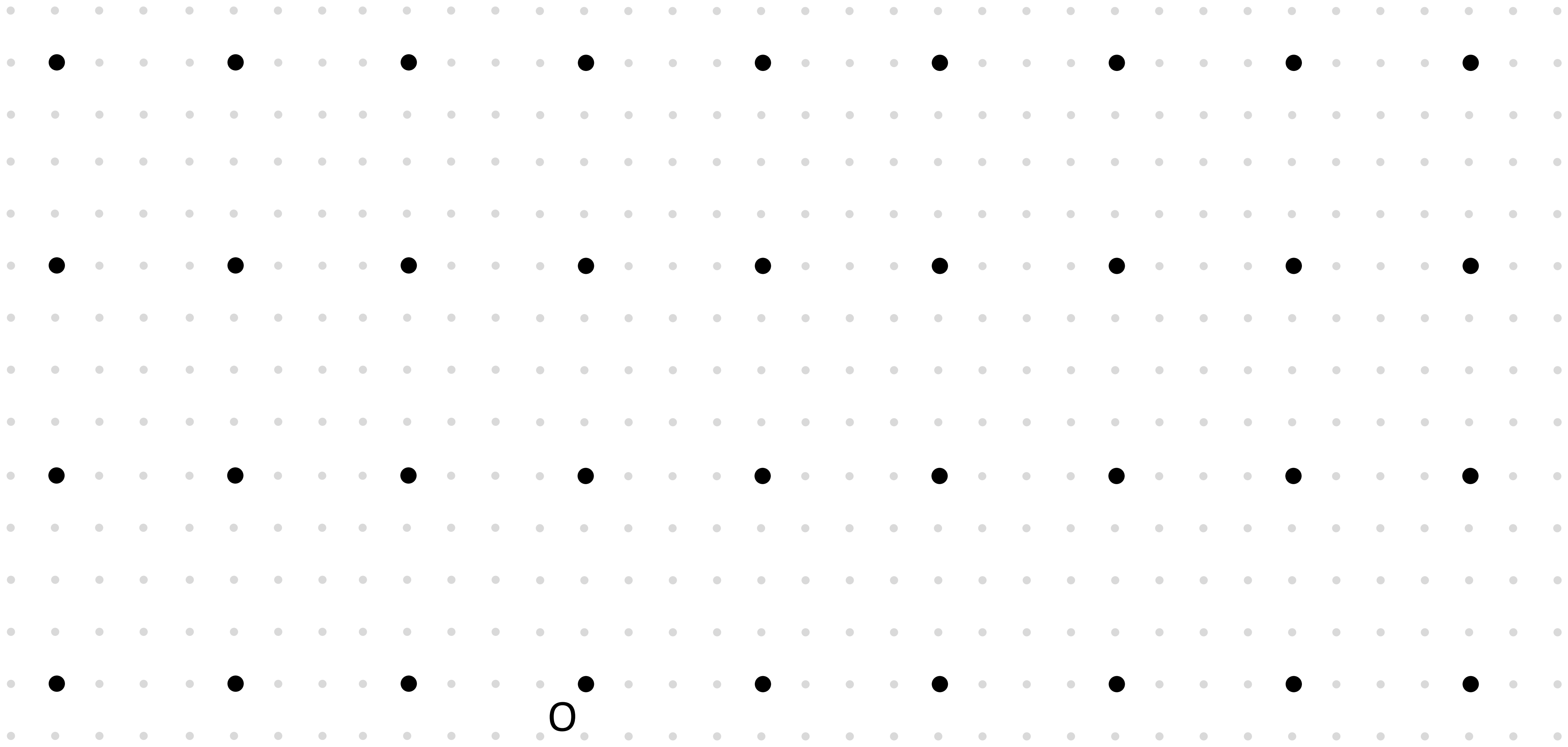
Homomorphic Encryption Framework (Multiplication)



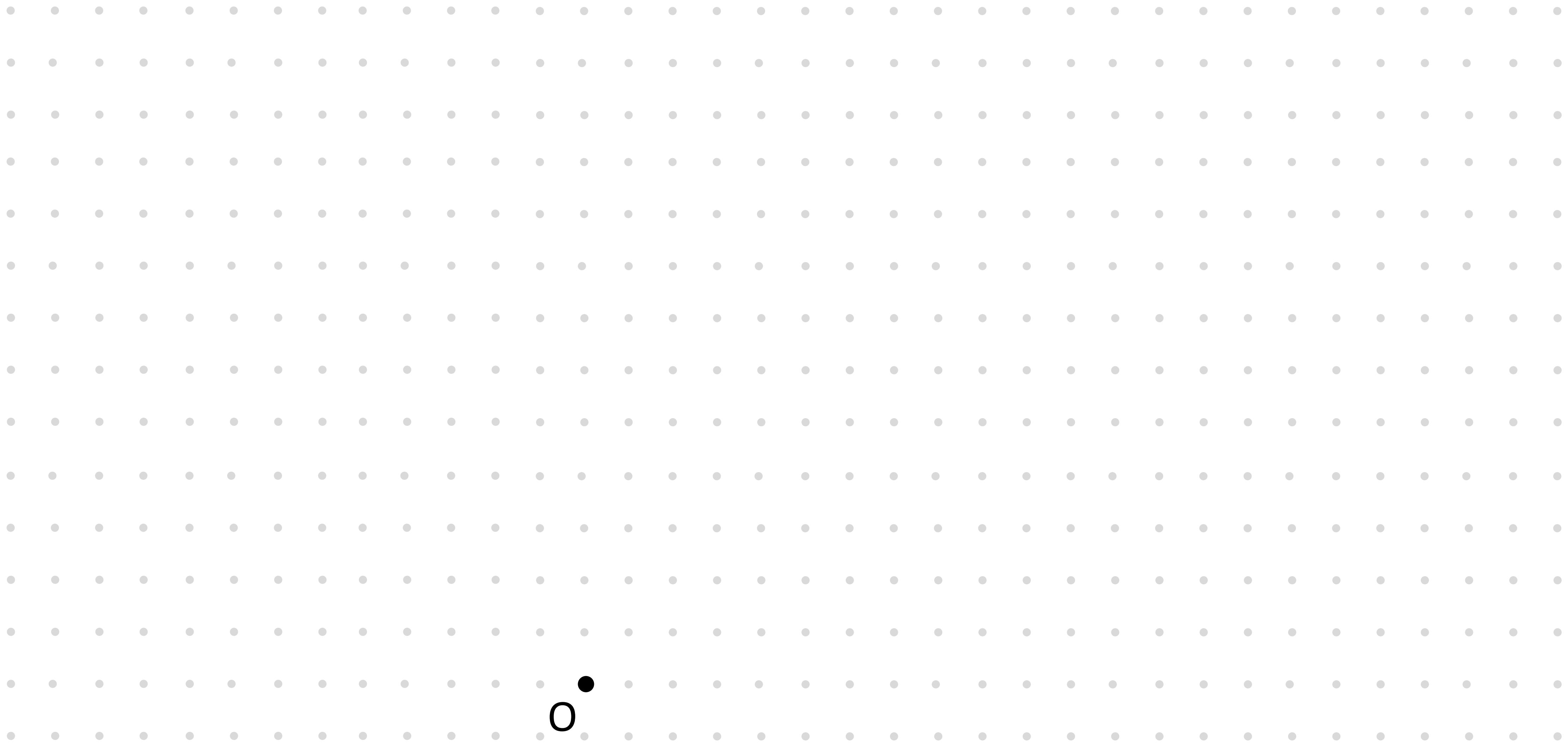
Homomorphic Encryption Framework (Multiplication)



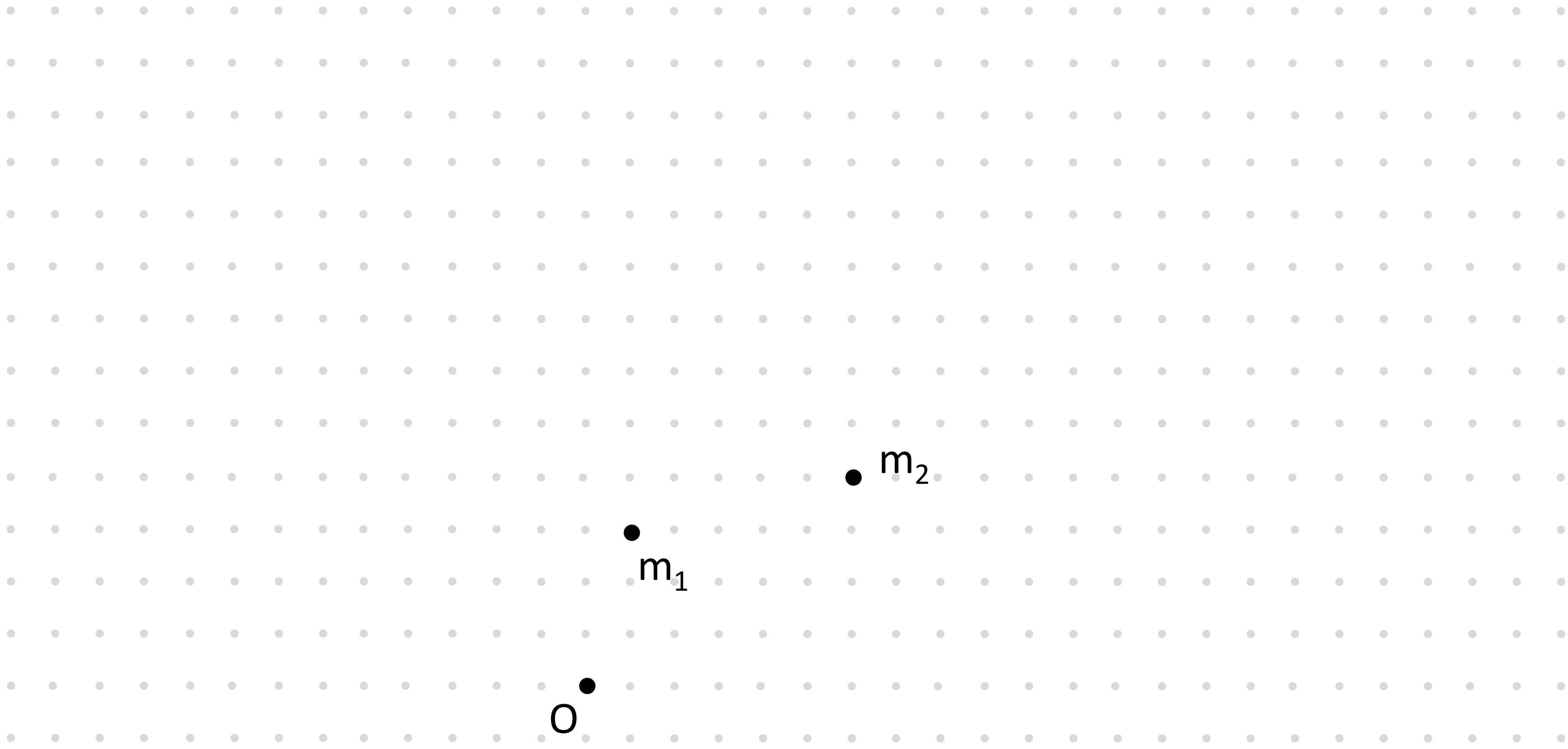
Approximate Homomorphic Encryption



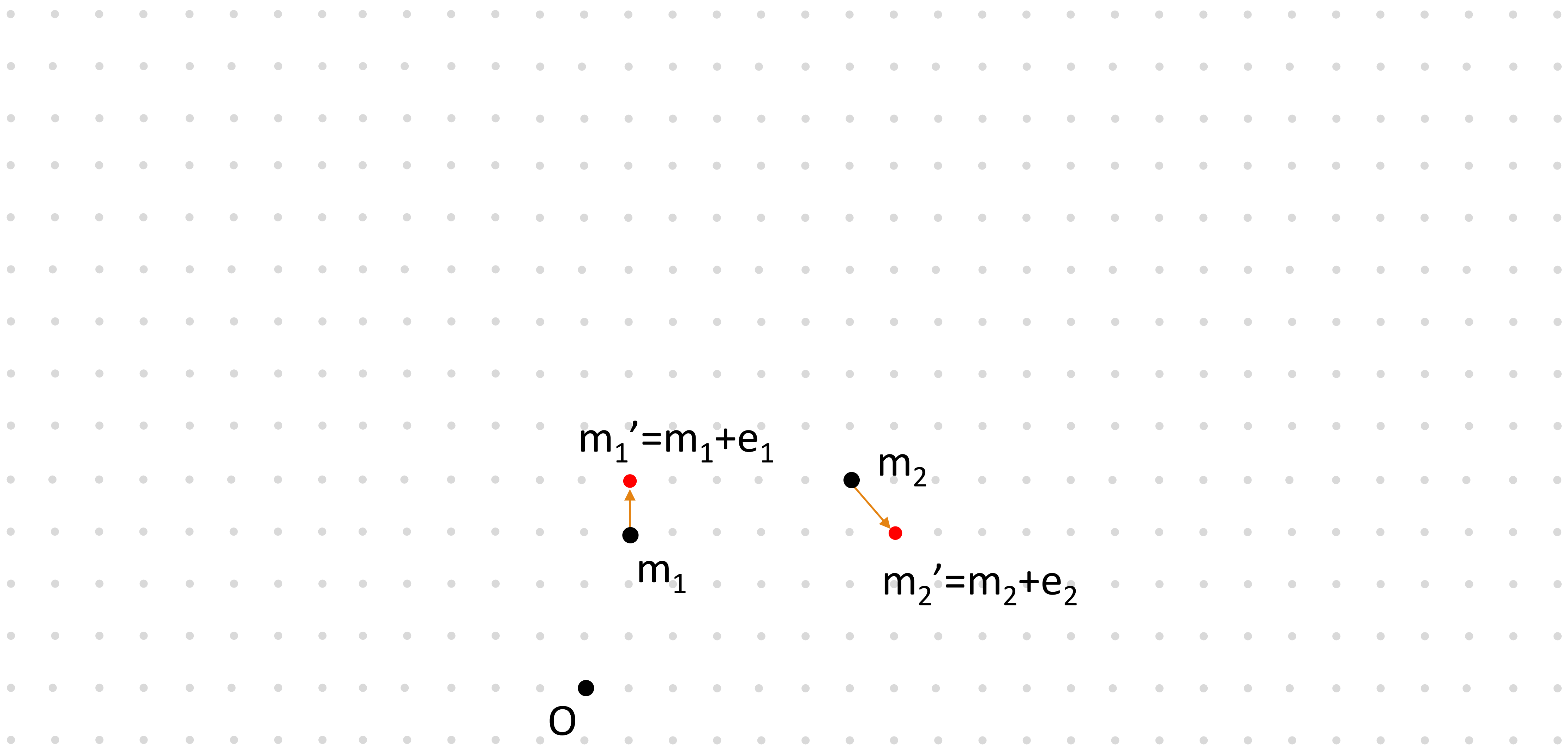
Approximate Homomorphic Encryption



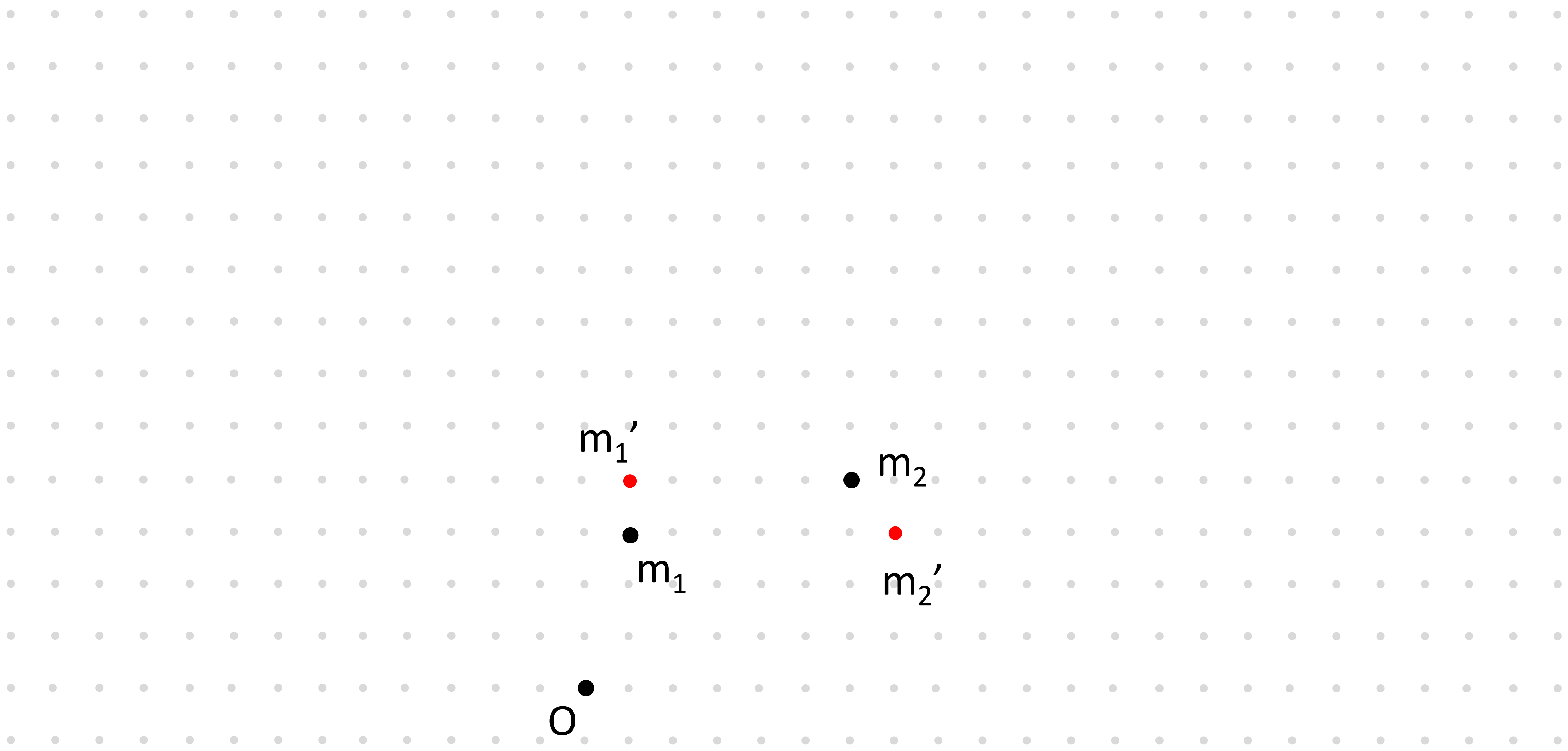
Approximate Homomorphic Encryption



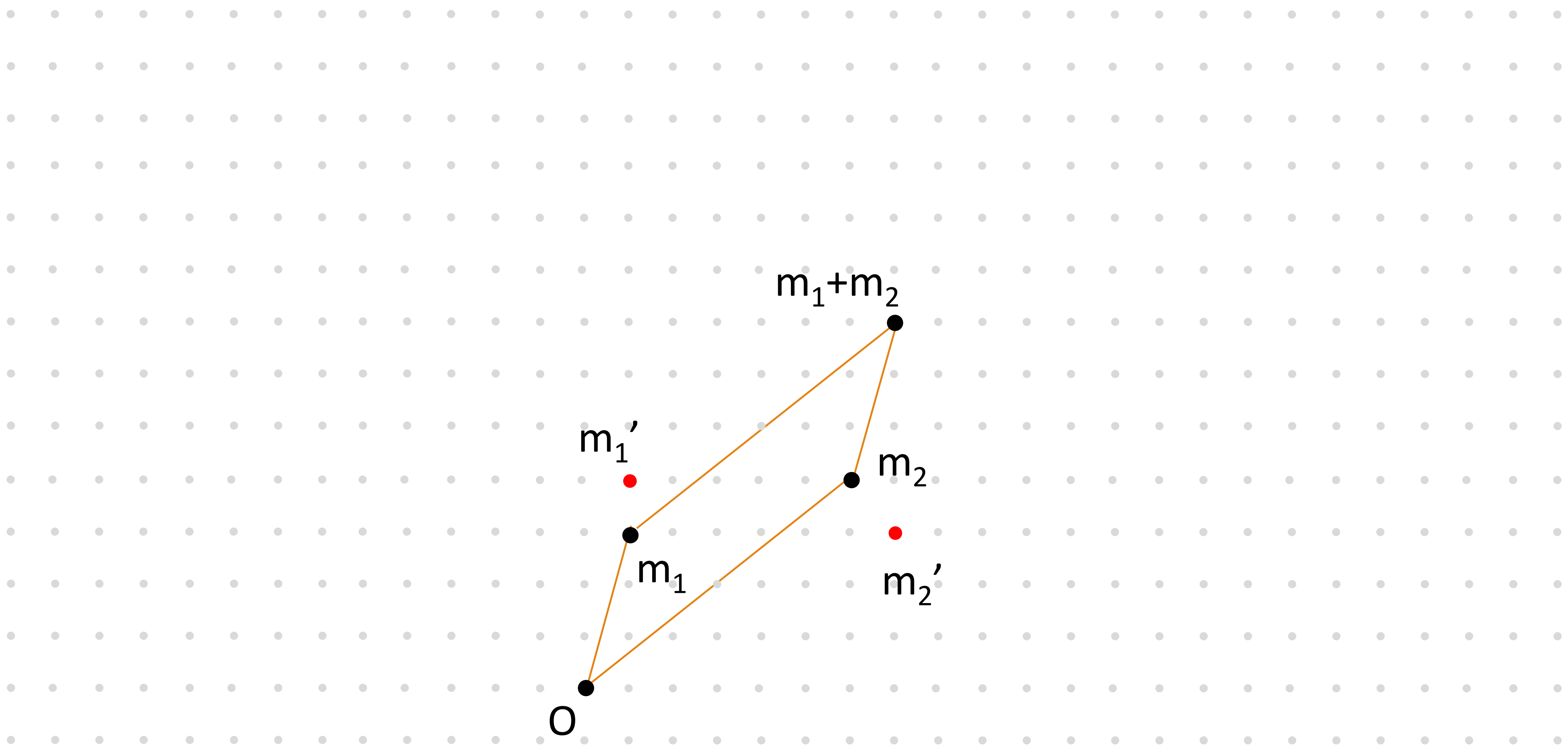
Approximate Homomorphic Encryption (Encryption)



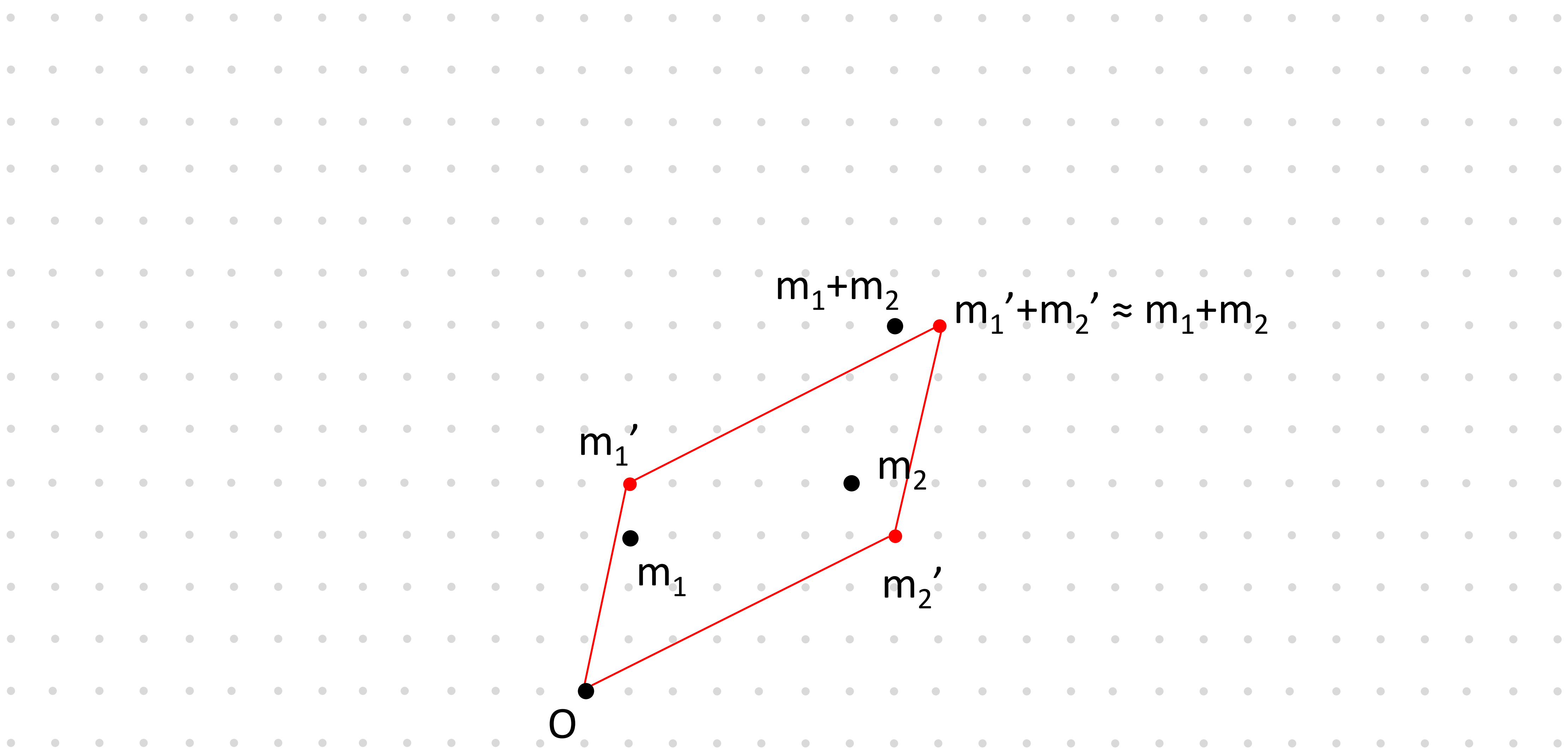
Approximate Homomorphic Encryption (Operations)



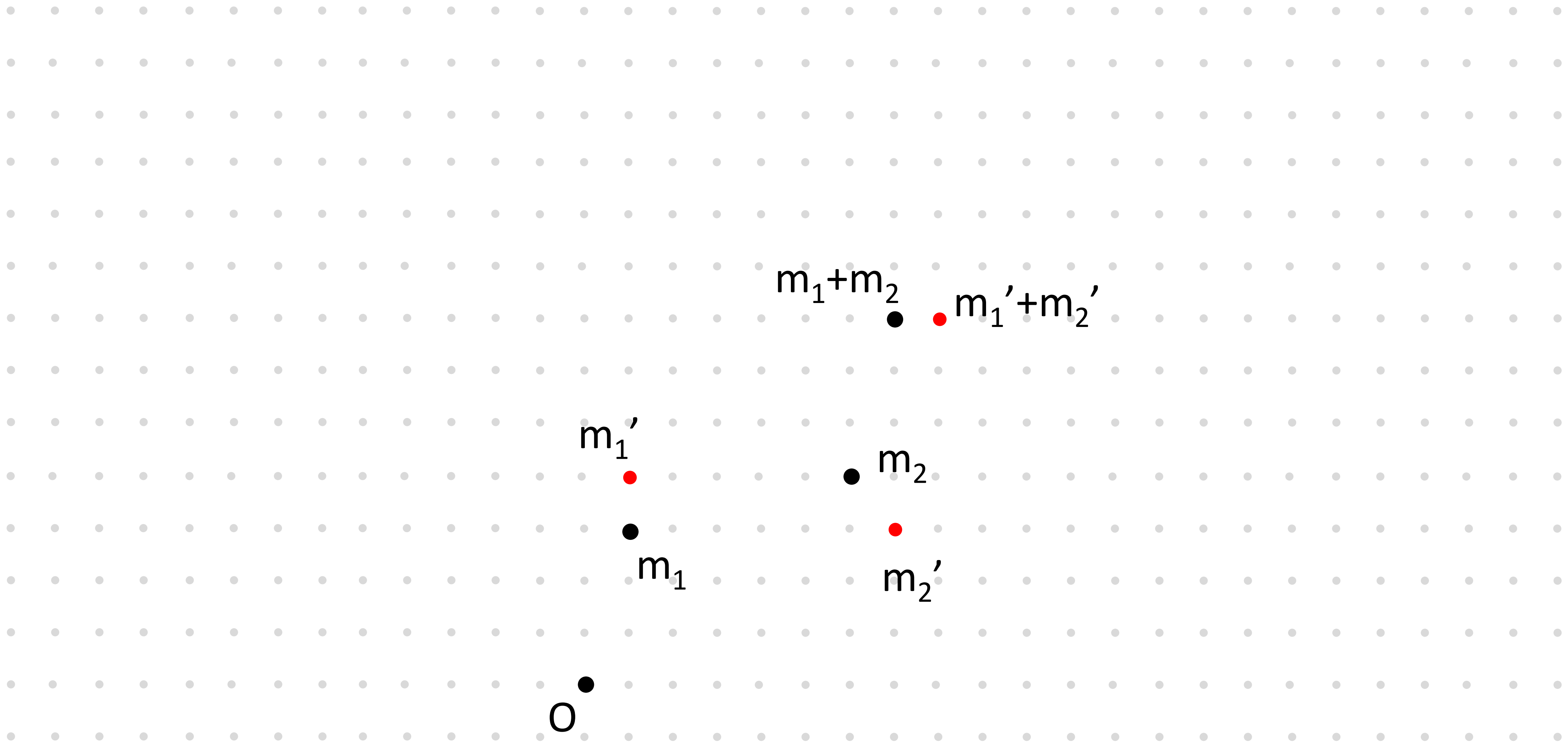
Approximate Homomorphic Encryption (Operations)



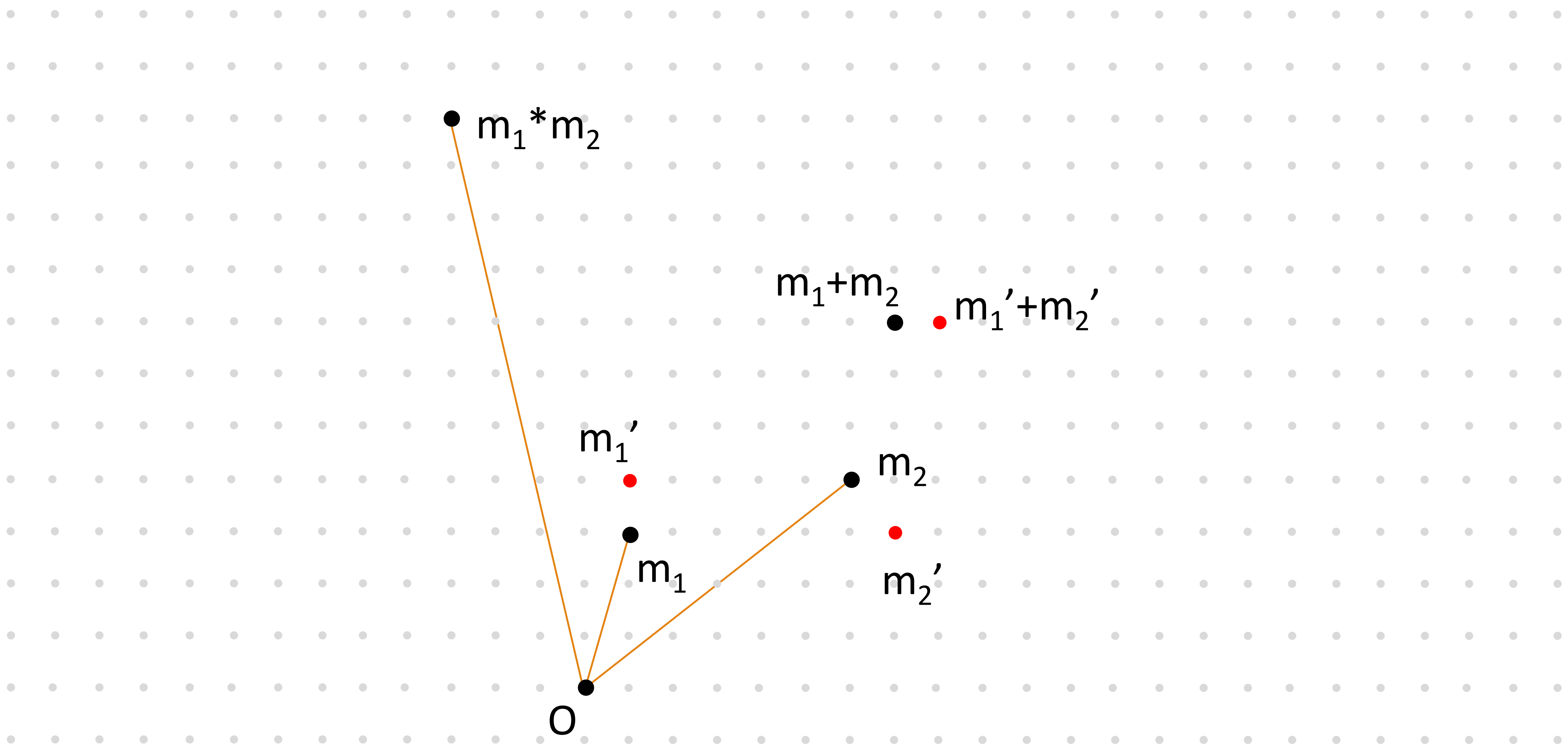
Approximate Homomorphic Encryption (Operations)



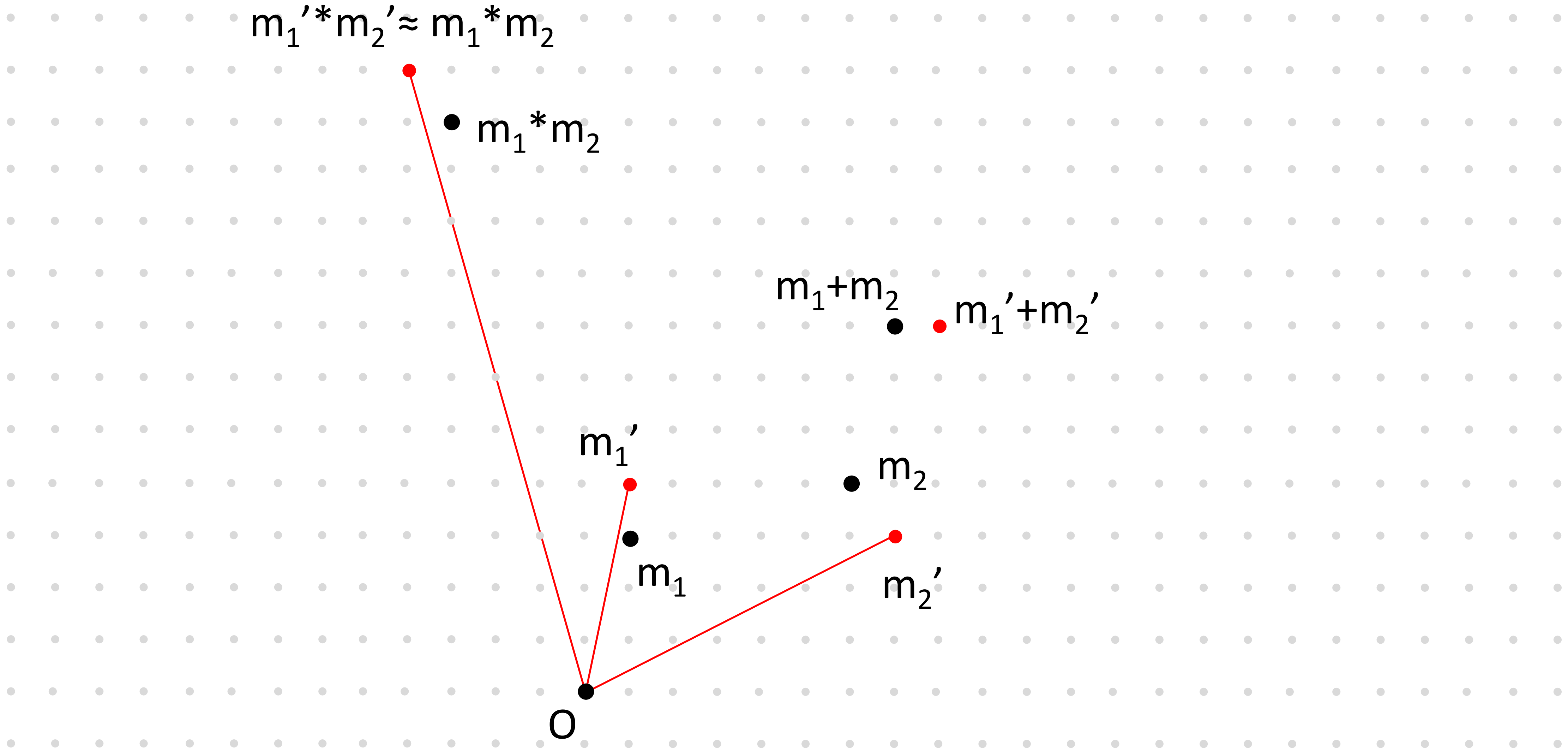
Approximate Homomorphic Encryption (Operations)



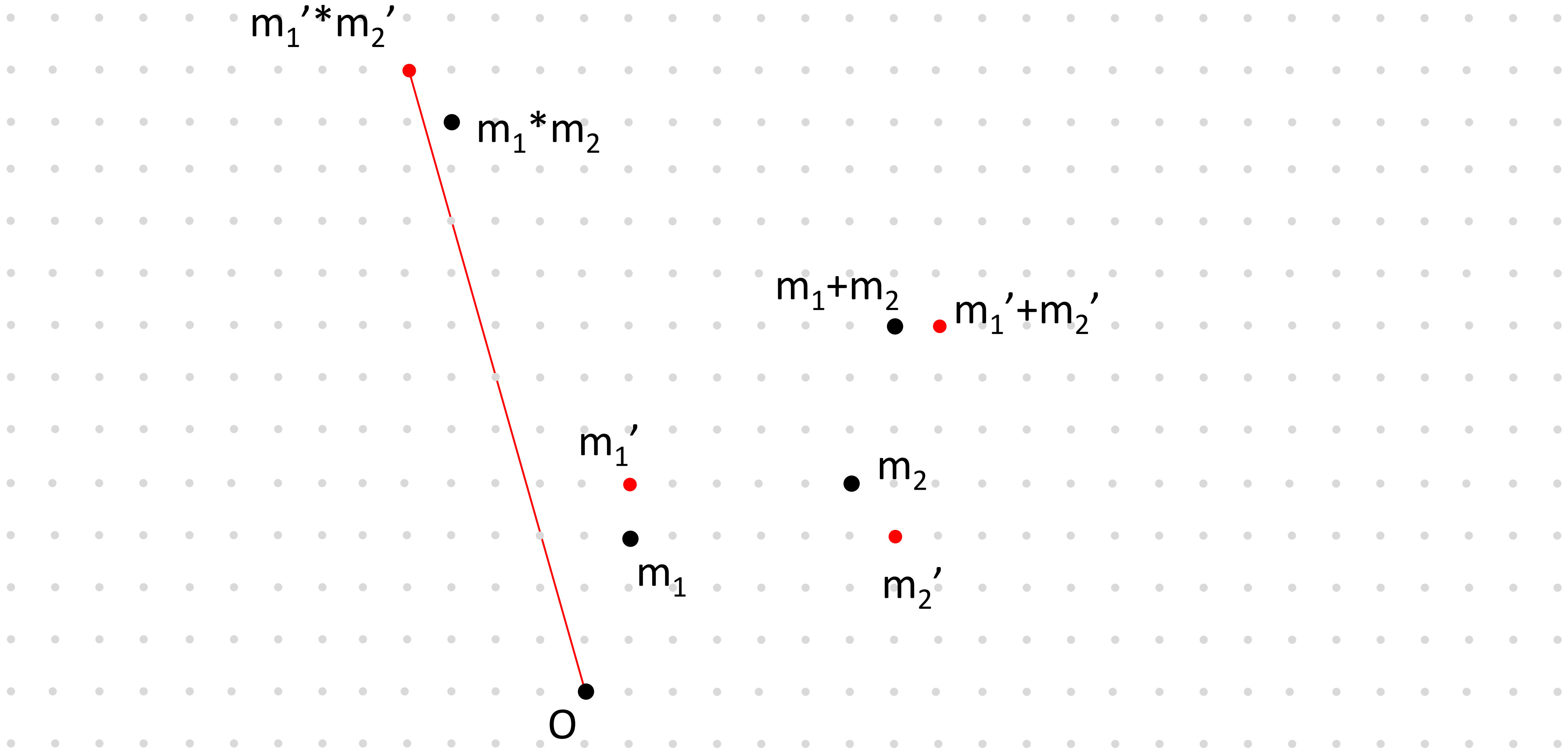
Approximate Homomorphic Encryption (Operations)



Approximate Homomorphic Encryption (Operations)



Approximate Homomorphic Encryption (Operations)



Approximate Homomorphic Encryption (Operations)