The Future of Cryptography: Performing Computations on Encrypted Data

Third-party cloud services reduce complexity and offer flexibility for enterprises. However, organizations need to be able to entrust their data—and their customers' data—to cloud service providers (CSPs) that are often incentivized to monetize these data. Meanwhile, regulations such as the US state of California's California Consumer Privacy Act (CCPA),[1] the US Consumer Online Privacy Rights Act (COPRA) bill[2] and the EU General Data Protection Regulation (GDPR)[3] aim to protect consumers' privacy, and noncompliant organizations are subjected to severe fines and suffer damaged reputations. This results in a tradeoff between data privacy and utility for organizations.[4] However, fully homomorphic encryption (FHE) allows organizations to ensure their customers' privacy without undermining their ability to gain insights from their data.

Homomorphic encryption allows for computations of encrypted data without the need to decrypt them. Instead, the resulting computations are preserved in an encrypted domain (consider plaintext to be the unencrypted domain and ciphertexts to be the encrypted domain), which, when decrypted, results in an output the same as if the operations were performed on an unencrypted domain. FHE can be used for privacy-preserving storage and computation. This allows data to be encrypted and outsourced to commercial cloud environments for processing while encrypted.[5]

Although there are many applications of FHE, consider two use cases: private contact discovery and log anomaly detection. For example, to add friends to a messaging service, users must upload their contact numbers or emails to the application's servers. Although the user's contacts may be encrypted for protection against eavesdropping during transmission to the application servers, the contacts must be decrypted on the servers to calculate hashes and detect any matches for others already using the service. The unencrypted data can be used in any manner, and users are forced to trust the application with their data. Another use case is detecting incidents of compromise (IoC) from log data. Third-party security tools such as security information and event management (SIEM) systems and extended detection and response (XDR) tools need access to unencrypted logs to detect anomalies. With FHE, these operations can be performed on encrypted data without compromising the privacy of the user's data.

Fully Homomorphic Encryption

The FHE scheme was first envisioned in 1978, within a year of the RSA scheme's publication.[6] Partially homomorphic encryption schemes supporting either addition or multiplication already existed, including:

- RSA, an asymmetric encryption used in online data transfers, which is based on the practical difficulty of factoring the product of two large prime numbers

DEVHARSH TRIVEDI

Is a Ph.D. candidate in cybersecurity at Stevens Institute of Technology (Hoboken, New Jersey, USA). He has worked as a senior software engineer at Philips and Oracle. He is a member of the Institute of Electrical and Electronics Engineers (IEEE) and enjoys volunteering at Positive Planet US as a chief information officer (CIO) and the ISACA New York, USA, Metropolitan Chapter as a data scientist. His publications are available at *https://scholar.google.com/citations?user=zxkRN_MAAAAJ&hl=en&oi=ao* and *https://www.researchgate.net/profile/Devharsh-Trivedi*.

- ElGamal (multiplicative homomorphism), an asymmetric key encryption algorithm based on the Diffie-Hellman key exchange for public-key cryptography, which provides a method of sharing a secret key, but does not allow secure communication
- Paillier (additive homomorphism), a probabilistic asymmetric algorithm for public key cryptography based on the problem that computing n^{th} residue classes are computationally intensive

US computer scientist Craig Gentry first proposed an FHE scheme based on lattices in 2009.[7] A lattice L(B) is the set of all integer combinations of the basis $B=\{b_1,\ldots, b_n\}$ of n linearly independent vectors. That is, a lattice is defined as:

$$L(B)=\{ B \cdot z : z \in Z^n\}$$

An FHE scheme supports both addition and multiplication (unlimited) operations, as illustrated by:

$$HE(a+b)=HE(a)+HE(b) \text{ and } HE(a*b)=HE(a)*HE(b)$$

There are three types of homomorphic encryption schemes:[8]

1. **Partially homomorphic encryption (PHE)**—Allows only a limited set of operations to be performed on encrypted data
2. **Somewhat homomorphic encryption (SHE)**—Allows a limited number of operations up to a certain complexity to be performed
3. **FHE**—Allows any mathematical operation to be performed an unlimited number of times

LOOKING FOR MORE?

- Learn more about, discuss and collaborate on information and cybersecurity in ISACA's Online Forums. *https://engage.isaca.org/onlineforums*

Use Cases

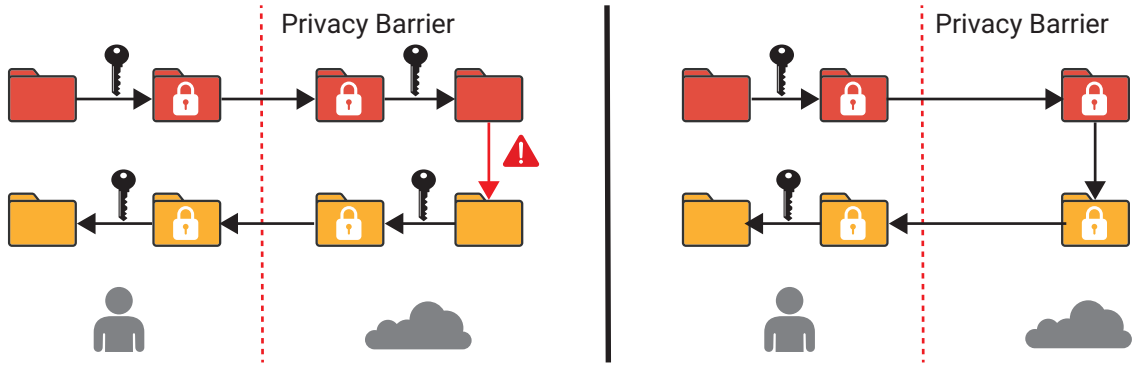
FHE can achieve privacy-preserving computation in many scenarios, including the application of private information retrieval (PIR) protocol, private search, private contact discovery, secure multiparty computation (MPC) and log anomaly detection in SIEM or XDR.[9] For example, the PIR protocol allows item retrieval from a server without revealing what was retrieved. MPC creates secure algorithms for involved users to jointly compute a function (process data) for their inputs while keeping those inputs private. In practice, Microsoft Edge applies FHE for password monitoring.[10]

With FHE, customer data privacy is ensured through cryptography, leveraging rigorous mathematical proofs.

As shown in **figure 1**, traditional (prevalent) cloud computation and storage solutions require customer data (by symmetric or asymmetric encryption schemes) to be decrypted before performing an operation (such as discovering how many of a user's contacts are using a messaging service) or detecting any anomalies from system logs. This exposes potentially sensitive customer data to cloud vendors. Customers must trust their provider's access control policies for data privacy.

FIGURE 1

Traditional Cloud Storage and Computation Model vs. FHE Model



With FHE, customer data privacy is ensured through cryptography, leveraging rigorous mathematical proofs. As a result, CSPs will not have access to unencrypted customer data for storage or computation.

Homomorphic Computations

Some FHE (both additive and multiplicative) include Brakerski-Gentry-Vaikuntanathan (BGV), Fan-Vercauteren (FV) or Brakerski-Fan-Vercauteren (BFV), and Cheon-Kim-Kim-Song (CKKS).[11] All these schemes are based on the hardness of the ring learning with errors (RLWE) problem, where noise is added during encryption and key generation to achieve hardness properties. To understand how computations are allowed on encrypted data, it is helpful to explore the underlying mathematics used to perform partial homomorphism in RSA, ElGamal and Paillier, and full homomorphism in BFV schemes.

RSA Cryptosystem (Unbounded Number of Modular Multiplications)

If the RSA public key has modulus n and encryption exponent e, then the encryption of a message m is given by $\mathcal{E}(m)=m^e \bmod n$.[12] The multiplication of two encrypted messages in RSA is:

$$\mathcal{E}(m_1) \cdot \mathcal{E}(m_2)=m_1^e m_2^e \bmod n$$
$$=(m_1 m_2)^e \bmod n$$
$$=\mathcal{E}(m_1 \cdot m_2)$$

As shown in **figure 2**, where the cells with darker (orange) backgrounds represent a correct result, adding two integers only yields an accurate result when the addition is zero or when adding any positive integer with a zero. In all other cases, it generates an incorrect summation.

Figure 3 shows the multiplicative nature of RSA, where it produces a correct result if the multiplication

FIGURE 2

Additive Operation in RSA

	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
-15	193	273	1055	328	318	697	856	975	17	331	1038	646	357	966	807	1106	911	786	1066	383	1056	998	898	337	792	913	1013	378	635	164	
-14	372	778	811	267	604	454	578	864	783	415	76	235	40	478	107	1107	299	941	234	258	37	583	681	319	758	616	237	422	567		957
-13	1055	811	242	1092	99	471	202	578	762	190	801	828	727	653	1118	1108	777	626	1	675	184	907	959	669	319	145	844	952		554	486
-12	328	267	1092	827	776	88	70	1006	114	511	946	382	421	536	71	1109	692	139	809	1069	134	332	371	282	753	336	740		169	699	743
-11	318	604	99	776	291	974	307	152	647	220	181	241	997	550	273	1110	458	302	449	660	433	593	681	747	1031	516		381	277	884	108
-10	697	454	471	88	974	876	380	966	158	197	239	968	1008	238	497	1111	308	337	407	1039	1024	528	392	252	895		605	785	976	505	208
-9	856	578	202	70	307	380	340	432	273	194	230	1089	847	632	205	1112	681	275	1082	1046	996	772	526	554		226	90	368	802	363	329
-8	975	864	578	1006	152	966	432	925	269	794	232	1088	425	1028	796	1113	278	913	732	595	313	502	75		567	869	374	839	452	802	784
-7	17	783	762	114	647	158	273	269	389	694	227	432	768	678	239	1114	1031	129	852	720	308	211		1046	595	729	440	750	162	360	223
-6	331	415	190	511	220	197	194	794	694	974	44	503	816	191	268	1115	630	1095	166	141	168		910	619	349	593	528	789	214	538	123
-5	1038	76	801	946	181	239	230	232	227	44	438	483	659	484	119	1116	729	1080	264	126		953	813	808	125	97	688	987	937	1084	65
-4	646	235	828	382	241	968	1089	1088	432	503	483	1023	397	544	514	1117	1017	858	251		995	980	401	526	75	82	461	52	446	863	738
-3	357	40	727	421	997	1008	847	425	768	816	659	397	487	812	656	1118	1108	631		870	857	955	269	389	39	714	672	312	1120	887	55
-2	966	478	653	536	550	238	632	1028	678	191	484	544	812	1072	272	1119	429		490	263	41	26	992	208	846	784	819	982	496	180	335
-1	807	107	1118	71	273	497	25	796	239	268	119	514	656	272	636	1120		692	13	104	392	491	90	843	440	813	663	429	344	822	210
0	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120																
1	911	299	777	692	458	308	681	278	1031	630	729	1017	1108	429			585	849	465	607	1002	853	882	325	916	624	848	1050	3	1014	314
2	786	941	626	139	302	337	275	913	129	1095	1080	858	631		692		849	49	309	577	637	930	443	93	489	883	571	585	468	643	155
3	1066	234	1	809	449	407	1082	732	852	166	264	251		490	13		465	309	634	724	462	305	353	696	274	113	124	700	394	1081	764
4	383	258	675	1069	660	1039	1046	595	720	141	126		870	263	104		607	577	724	98	638	618	689	33	32	153	880	739	293	886	475
5	1056	37	184	134	433	1024	996	313	308	168		995	857	41	392		1002	637	462	638	683	1077	894	889	891	882	940	175	320	1045	83
6	998	583	907	332	593	528	772	502	211		953	980	955	26	491		853	930	305	618	1077	147	427	327	927	924	901	610	931	706	790
7	898	761	959	371	681	392	526	75		910	813	401	269	992	90		882	443	353	689	894	427	732	852	848	963	474	1007	359	338	1104
8	337	319	669	282	747	252	554		1046	619	808	526	389	208	843		325	93	696	33	889	327	852	196	689	155	969	115	543	257	146
9	792	758	319	753	1031	895		567	595	349	125	75	39	846	440		916	489	274	32	891	927	848	689	781	741	814	1051	919	543	265
10	913	616	145	336	516		226	869	729	593	97	82	714	784	813		624	883	113	153	882	924	963	155	741	245	147	1033	650	667	424
11	1013	237	844	740		605	90	374	440	528	688	461	672	819	663		848	571	124	880	940	901	474	969	814	147	830	345	1022	517	803
12	378	422	952		381	785	368	839	750	789	987	52	312	982	429		1050	585	700	739	175	610	1007	115	1051	1033	345	294	29	854	793
13	635	567		169	277	976	802	452	162	214	937	446	1120	495	344		3	468	394	293	320	931	359	543	919	650	1022	29	879	310	66
14	164		554	699	884	505	363	802	360	538	1084	863	887	180	822		1014	643	1081	886	1045	706	338	257	543	667	517	854	310	343	649
15		957	486	743	108	208	329	784	223	123	65	738	55	335	210		314	155	764	475	83	790	1104	146	265	424	803	793	66	749	928

FIGURE 3

Multiplicative Operation in RSA

	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
-15																	1106	1091	1076	1061	1046	1031	1016	1001	986	971	956	941	926	911	896
-14																	1107	1093	1079	1065	1051	1037	1023	1009	995	981	967	953	939	925	911
-13																	1108	1095	1082	1069	1056	1043	1030	1017	1004	991	978	965	952	939	926
-12																	1109	1097	1085	1073	1061	1049	1037	1025	1013	1001	989	977	965	953	941
-11																	1110	1099	1088	1077	1066	1055	1044	1033	1022	1011	1000	989	978	967	956
-10																	1111	1101	1091	1081	1071	1061	1051	1041	1031	1021	1011	1001	991	981	971
-9																	1112	1103	1094	1085	1076	1067	1058	1049	1040	1031	1022	1013	1004	995	986
-8																	1113	1105	1097	1089	1081	1073	1065	1057	1049	1041	1033	1025	1017	1009	1001
-7																	1114	1107	1100	1093	1086	1079	1072	1065	1058	1051	1044	1037	1030	1023	1016
-6																	1115	1109	1103	1097	1091	1085	1079	1073	1067	1061	1055	1049	1043	1037	1031
-5																	1116	1111	1106	1101	1096	1091	1086	1081	1076	1071	1066	1061	1056	1051	1046
-4																	1117	1113	1109	1105	1101	1097	1093	1089	1085	1081	1077	1073	1069	1065	1061
-3																	1118	1115	1112	1109	1106	1103	1100	1097	1094	1091	1088	1085	1082	1079	1076
-2																	1119	1117	1115	1113	1111	1109	1107	1105	1103	1101	1099	1097	1095	1093	1091
-1																	1120	1119	1118	1117	1116	1115	1114	1113	1112	1111	1110	1109	1108	1107	1106
0																															
1	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120																
2	1091	1093	1095	1097	1099	1101	1103	1105	1107	1109	1111	1113	1115	1117	1119																
3	1076	1079	1082	1085	1088	1091	1094	1097	1100	1103	1106	1109	1112	1115	1118																
4	1061	1065	1069	1073	1077	1081	1085	1089	1093	1097	1101	1105	1109	1113	1117																
5	1046	1051	1056	1061	1066	1071	1076	1081	1086	1091	1096	1101	1106	1111	1116																
6	1031	1037	1043	1049	1055	1061	1067	1073	1079	1085	1091	1097	1103	1109	1115																
7	1016	1023	1030	1037	1044	1051	1058	1065	1072	1079	1086	1093	1100	1107	1114																
8	1001	1009	1017	1025	1033	1041	1049	1057	1065	1073	1081	1089	1097	1105	1113																
9	968	995	1004	1013	1022	1031	1040	1049	1058	1067	1076	1085	1094	1103	1112																
10	971	981	991	1001	1011	1021	1031	1041	1051	1061	1071	1081	1091	1101	1111																
11	956	967	978	989	1000	1011	1022	1033	1044	1055	1066	1077	1088	1099	1110																
12	941	953	965	977	989	1001	1013	1025	1037	1049	1061	1073	1085	1097	1109																
13	926	939	952	965	978	991	1004	1017	1030	1043	1056	1069	1082	1095	1108																
14	911	925	939	953	967	981	995	1009	1023	1037	1051	1065	1079	1093	1107																
15	896	911	926	941	956	971	986	1001	1016	1031	1046	1061	1076	1091	1106																

is nonnegative and an inaccurate result if the multiplication is negative.

ElGamal Cryptosystem (Unbounded Number of Modular Multiplications)

In the ElGamal cryptosystem, in a cyclic group G of order q with generator g, if the public key is (G, q, g, h), where $h=g^x$ and x is the secret key, then the encryption of a message m is $\mathcal{E}(m)=(g^r, m \cdot h^r)$, for some random r $\mathcal{E}\{0,..., q\text{-}1\}$.[13] Multiplication of two ciphers in ElGamal is:

$$\mathcal{E}(m_1) \cdot \mathcal{E}(m_2)=(g^{r_1}, m_1 \cdot h^{r_1}) (g^{r_2}, m_2 \cdot h^{r_2})$$
$$=(g^{r_1+r_2}, (m_1 \cdot m_2) h^{r_1+r_2})$$
$$=\mathcal{E}(m_1 \cdot m_2)$$

As shown in **figure 4**, ElGamal does not predictably produce a correct result for adding two integers. However, **figure 5** shows the multiplicative nature of ElGamal, where it generates an accurate result for both negative and nonnegative multiplications.

Paillier Cryptosystem (Unbounded Number of Modular Additions)

In the Paillier cryptosystem, if the public key is the modulus n and the base g, then the encryption of a message m is $\mathcal{E}(m)=g^m r^n \mod n^2$, for some random $r \mathcal{E} \{0,..., n\text{-}1\}$.[14] The addition of two encrypted messages in Pallier is:

$$\mathcal{E}(m_1) \cdot \mathcal{E}(m_2)=(g^{m_1} r_1^n)(g^{m_2} r_2^n) \mod n^2$$
$$=g^{m_1+m_2} (r_1 r_2)^n \mod n^2$$
$$=\mathcal{E}(m_1+m_2)$$

FIGURE 4

Additive Operation in ElGamal

	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
-15	-23	-17	-45	-15	-38	-28	-34	-36	-20	-17	-24	-16	-17	-15			-13		-14	164	-9	-12	2	0	-8	4	-10	19	15	-8	2
-14	-14	-43	-126	-29	-14	-27	-71	-27	-152	-15			-16	-18	-14		-5	-8	10	-12	-6	-7	-12	55	2	-7	-20	-13	16	-5	-11
-13	-17	-13	-46	-100	-22	-45	-19	-31	-13	-24	-174	-32	-21	-13			-10	5		42	-11	-11	-4	-6	-11	-9	-6	-9	557		5
-12	-20	-28	-34	-17	-52	-87	-22	-15	-30	-21	-18	-13	-25	-13	-14			-9	-10	-10	-10	17	2	9	45	34	6	-2	0	33	109
-11	-12	-207	-17	-39	-198	-18	-26		-1908		-14	-12	-37	-21	-372		0	4	-7	-8	-2			701	-7	-8	-6	-8	0	45	
-10	-16	-21	-10		-22	-17	-559	-16	-10	-24	-16	-26	-22	-248	-13		-8	-6	-3	61	-9		-6	14	9	-9	3	158	-6	-6	-8
-9	-16	-32	-4117	-28	-11	-481	-11	-14	-17	-9	-17	-31	-11	-9	-9			-8	-7	2627	-1	-4			371	9	10	6483	2149	3	4
-8	-13	-36		-22	-8	-44	-8	-13	-21	-17	-19	-24			-8			07	-7	-3		2866	-6	-5	-4	0	4	-6	27	22	19
-7	-15	-14	-7	-14	-23	-10	-104	-7	-10	-7	-10	-18	-66	-10	-24				-6	-6	-4	-4		-6	3		-6	386	0	20	5
-6	-16	-52	-34	-7	-18	-9	-21		-10	-30	-10	-17	-6	-9	-6		-4				-3	-5	5	-2	0	19		-2	15	24	416
-5	-30	-10	-22	-24	-22	-49	-20	-9	-6	-6		-12	-5	-6	-5		-3	-2	0	0	-2	-4	30	0	-2	2	19	101	9	0	3
-4	-15	-34	-20	-29	-14	-9	-10	-21	-5	-31	-7	-5	-4	-80	-4				0	-1	-2		12	8	3632	26	1	2	5	7990	21
-3	-271	-106	-1160	-16	-3	-1063	-27	-107		-15	-7	-5		-4				-2	14	14	7	20	10	0	0	21	0	26	2	17	-1
-2	-565	-2	-4	-6	-15	-11	-12	-7	-15	-3	-1732	-19	-2	-2	-318		0		3	0	2	3	35		2	4	22	30	-1	4	-1
-1	-11		-1	-31	0-17	-39	-5	-26	-59	-445	-14	-13	-1	-1	-1		3	5	3	0	1	53	9	0	18	61	2	0	4	23	19
0	-4	-92	-98	-8	-7	-20	-144	-47	-5	-35	-4	0	0	-7			0	1	0	5	4	13			88	0	1	6	0	235	69
1	-34	-12	-3	-2	-11	-18	-26	0	-14	-6	-3	-1	-716				1	5	6	2	2	3	3	1	20	12	2	21	266		13
2	-119	-1	-2	-14	-5	0	-2	-3950	1	-9	-10	0	-11				2	67	20		2		22	4	313	3612		34	14	5	5
3	-2	-28	-2	-10	-118	0	2	1	-8	0	0	0	-1	0	-7		6	9	7	22	5	17	9		5664	21	8	9		14	6
4	0		-7	0	-4	-25	-23	-14	-12	2	-30	1	-137	3				7	6	5	16	8	5	4	12	5	8	1	7	11	65
5	-34	-7	-23	4	-89	4	-46	-5	-13	2	1	-1	4	-7	3		5	5	6	34		12	10	7	71	17	11	16	14	26	11
6	3	5	3	0	-3	3	5	-3	-15	-5	-3	3		-11	4		6	7		6		8	6	6	10	10	2	17	11	9	13
7	-5	6	6	-6	-3	-6	-9	-7	5	6	-4	-49	0				7	8	12	10	25	1453	145	23	7	14	19	16	8	16	7
8	1	-77	-24	-13		5	1	4	0	0	-3297	-2	7	5			8	9	950	10		11		8	11	16	35	11	11	8	15
9	2	8	6	-54	-345	4	5	6	-11	6		-19	8	8			9	15		10	12	17	31	18	57	308	491	20		43	115
10	-64	1	-49	2	0		5	4		7	6	7	0		8		13	10	14	11	16		298	45	16	3000	24	18	28	74	23
11	10	-5	7	-28	-6		7	8			8	5	10	8			11	12		23	119	13	19	28	12	25	25		13	14	30
12	-1	3	5	-13	-1484	11			6	10	-5	2			-2		24			71	15	12	31	16	30	12	21	78	13	49	
13	12	-2		6	-56		8		4	9	12	12	-395	4			13	28	40		16	23	41	17	16	41	17	15	123	23	167
14	-69	7	13	12	12	2	9	13	0	7	10		7				14		121	15	16	740	4641	25	165	40	52	23	32	43	23
15	-39		10	1	0	6	-21	-16	1	12	13	0		12	10		17		16	15	24	26	39	22	816	82	28	18	82	10	20

Fan-Vercauteren (FV) or Brakerski-Fan-Vercauteren (BFV) Scheme

FV/BFV and BGV schemes are very similar, and the computations are performed on integers. However, in CKKS, calculations can be performed on complex numbers with limited precision.[15] This implies that BFV and BGV are better choices to obtain accurate results, and CKKS is best suited for machine learning (ML) tasks as results in CKKS are approximated values.

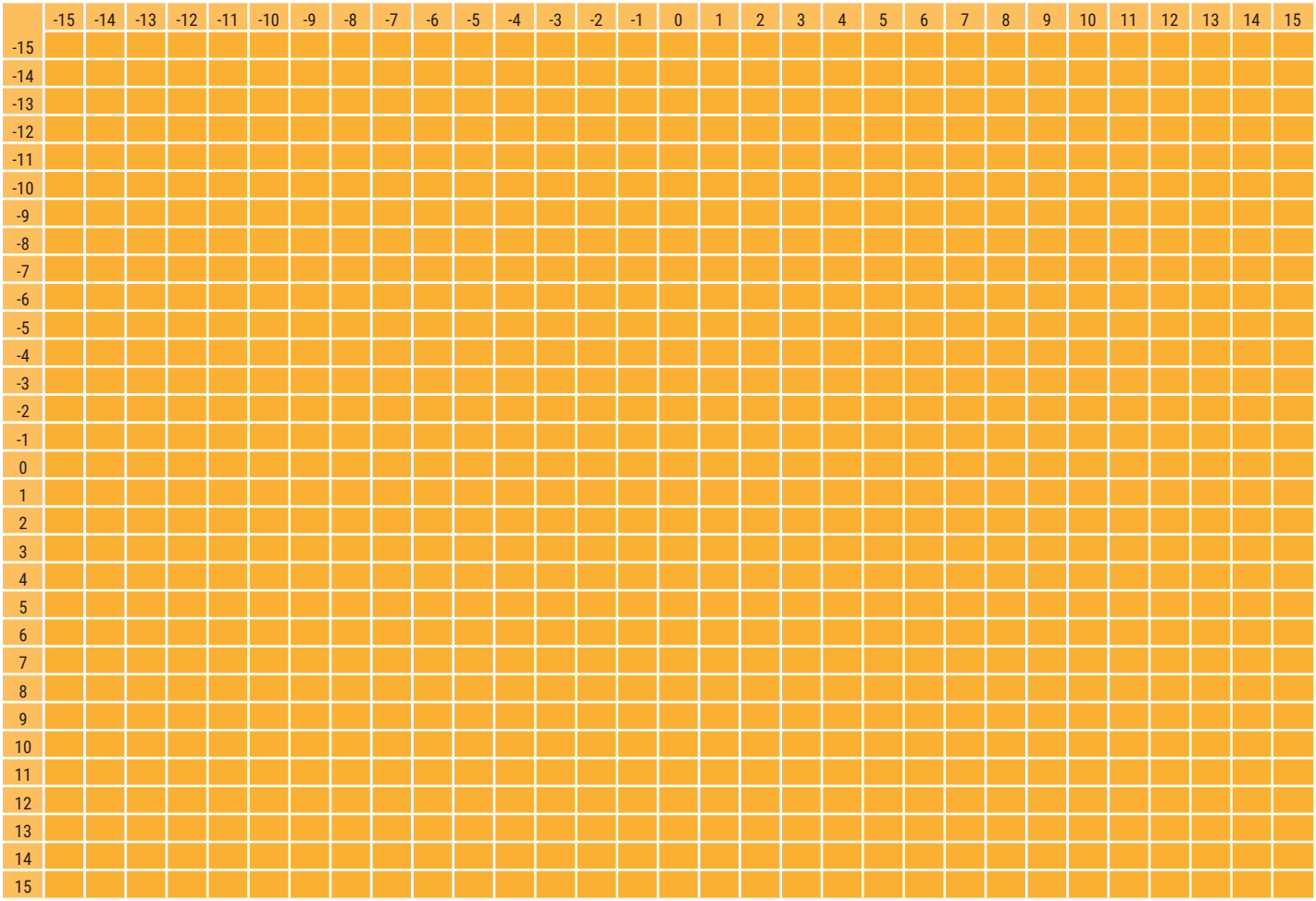
BFV and CKKS allow batching to put a plaintext vector (batch) inside a single ciphertext. These so-called batched schemes pack multiple values into a single ciphertext (typically thousands) and perform operations on all the values for the cost of a single homomorphic operation. Batching is one of the more prominent sources of speedup since the discovery of FHE. CKKS is especially good for numeric and ML applications because the approximation it implies can be managed, and it uses faster encryption parameters than BGV and BFV.

To encrypt a plaintext message M in the plaintext domain P:[16]

- Generate a random polynomial u from R_2, where R_2 is the key distribution used to generate polynomials with integer coefficients -1,0 or 1.
- Generate two small random polynomials, e_1 and e_2 from *x*, where x is the error distribution (usually a discrete Gaussian distribution) defined with parameters mean μ and standard deviation σ over R bounded by some integer β. e_1 and e_2 are referred to as error or noise terms.

FIGURE 5

Multiplicative Operation in ElGamal



- A ciphertext C for a message M is a pair of values C_1 and C_2. $C=(C_1,C_2)$ in encrypted domain C can be described as:

$$C_1=[PK_1 \cdot u+ e_1+ \Delta M]_q$$
$$C_2=[PK_2 \cdot u+ e_2]_q$$

R_q is a uniform random distribution over R_q. The notation $[\cdot]_q$ means that polynomial arithmetic should be done modulo q.[17]

For reference, homomorphic addition H_+ for two ciphertexts is:

$$H_+ (C^{(1)} ,C^{(2)})=([C_1^{(1)}+C_1^{(2)}]_q,[C_2^{(1)}+C_2^{(2)}]_q)= (C_1^{(3)},C_2^{(3)}$$
$$=([PK_1 \cdot (u^{(1)}+u^{(2)})+(e_1^{(1)}+e_1^{(2)})+ \Delta(M^{(1)}+M^{(2)}])_q,$$
$$([PK_2 \cdot (u^{(1)}+u^{(2)})+(e_2^{(1)}+e_2^{(2)})]_q$$
$$= ([PK_1 \cdot u^{(3)}+e_1^{(3)}+ \Delta(M^{(1)}+M^{(2)})]_q, ([PK_2 \cdot u^{(3)}+e_2^{(3)}]_q$$

Adding error to ciphers provides security but introduces limitations for multiplications as the error grows with each multiplication of ciphers. If the error grows large enough, the cipher can no longer be decrypted successfully.

Conclusion

FHE is the emerging champion of cryptography with potential use cases for PIR, MPC, private contact discovery and privacy-preserving log anomaly detection. FHE allows computations on encrypted data without decryption, which can help organizations comply with privacy regulations. The mathematical operations of partially homomorphic schemes—such as RSA, ElGamal and Paillier—and fully homomorphic BFV schemes are helpful for practitioners who wish to dive deep into FHE and incorporate it into their security projects. Tech

giants are rooting for fully homomorphic encryption schemes (e.g., IBM Security offers fully homomorphic encryption as a service).[18, 19] However, current computation limitations regarding the types of operations allowed and computation time remain a hurdle for immediate adaption.

Endnotes

1 California Privacy Protection Agency (CPPA), "California Consumer Privacy Act Regulations," USA, *https://cppa.ca.gov/regulations/consumer_privacy_act.html*

2 US Senate, S.3195 Consumer Online Privacy Rights Act, 117th Congress, USA, 2021, *https://www.congress.gov/bill/117th-congress/senate-bill/3195*

3 GDPR, "Complete Guide to GDPR Compliance," *https://gdpr.eu/*

4 Dilmegani, C.; "What Is Homomorphic Encryption? Benefits and Challenges," AI Multiple, 12 October 2022, *https://research.aimultiple.com/homomorphic-encryption/*

5 Zagakos, A.; "What Is Homomorphic Encryption?" FreeCodeCamp, 26 April 2022, *https://www.freecodecamp.org/news/introduction-to-homomorphic-encryption/*

6 Rivest, R. L.; L. Adleman; M. L. Dertouzos; "On Data Banks and Privacy Homomorphisms," Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 1978, *https://people.csail.mit.edu/rivest/RivestAdlemanDertouzos-OnDataBanksAndPrivacyHomomorphisms.pdf*

7 Gentry, C.; "Fully Homomorphic Encryption Using Ideal Lattices," Proceedings of the 41st Annual ACM Symposium on Theory of Computing, May 2009, *https://dl.acm.org/doi/10.1145/1536414.1536440*

8 *Op cit* Dilmegani

9 Bhattacharya, A.; "Homomorphic Encryption—Basics," Encryption Consulting, 24 December 2020, *https://www.encryptionconsulting.com/introduction-to-homomorphic-encryption/*

10 Lauter, K.; S. Kannepalli; K. Laine; R. Cruz Moreno; "Password Monitor: Safeguarding Passwords in Microsoft Edge," Microsoft Research Blog, 21 January 2021, *https://www.microsoft.com/en-us/research/blog/password-monitor-safeguarding-passwords-in-microsoft-edge/*

11 Thaine, P.; "Homomorphic Encryption for Beginners: A Practical Guide (Part 1)," *Medium*, 26 December 2018, *https://medium.com/privacy-preserving-natural-language-processing/homomorphic-encryption-for-beginners-a-practical-guide-part-1-b8f26d03a98a*

12 Vadhan, S.; A. Rosen; "Public-Key Encryption in Practice," Introduction to Cryptography, Harvard John A. Paulson School of Engineering and Applied Sciences, Cambridge, Massachusetts, USA, 16 November 2006, *https://people.seas.harvard.edu/~salil/cs120/docs/lec15.pdf*

13 Chen, N.; "A Comparison of El Gamal and Paillier Cryptosystems," University of California Santa Barbara, California, USA, June 2018, *http://koclab.cs.ucsb.edu/teaching/cren/project/2018/Chen.pdf*

14 Mohammed, S. J.; D. B. Taha; "Performance Evaluation of RSA, ElGamal, and Paillier Partial Homomorphic Encryption Algorithms," IEEE 2022 International Conference on Computer Science and Software Engineering (CSASE), Mach 2022

15 *Op cit* Thaine

16 Inferati Inc., *Introduction to the BFV FHE Scheme*, USA, *https://inferati.azureedge.net/docs/inferati-fhe-bfv.pdf*

17 *Ibid*.

18 IBM, "Homomorphic Encryption Services," *https://www.ibm.com/security/services/homomorphic-encryption*

19 Eatwell, A.; "Intel, Microsoft Push Homomorphic Encryption With Open-Source Moves," Spiceworks, 10 January 2019, *https://www.spiceworks.com/tech/artificial-intelligence/articles/intel-microsoft-push-homomorphic-encryption-with-open-source-moves/*