

¡Correo Rojo! Volamos hacia Madrid

Ana I. Gómez Pérez
ana.gomez.perez@urjc.es

12 de febrero de 2026

1. El criptosistema de Vernam

En este reto os proponemos estudiar la seguridad del *criptosistema de Vernam* o también conocido como el *one-time pad* (el cuaderno de un solo uso), bajo la reutilización de claves criptográficas. Este es un algoritmo clásico con la siguiente definición: Para un texto de entrada de longitud N , el texto cifrado se genera según la siguiente ecuación:

$$C_j = P_j \oplus K_j, \quad j = 1, \dots, N$$

donde K es la clave secreta compartida. Dada la salida y la clave, aplicando el mismo procedimiento obtenemos la entrada. Se considera *incondicionalmente seguro*, siempre que K sea de la misma longitud que P y que K tenga estructura completamente aleatoria

El criptosistema de Vernam fue el primer criptosistema que se supo que era *incondicionalmente seguro*, es decir, es un algoritmo donde solamente se puede obtener la verdadera entrada si se conoce la clave, ya que conocer la salida no da información adicional sobre la entrada si todas las posibles claves tienen la misma probabilidad. Este criptosistema se utilizó durante la guerra fría para poder comunicar a los gobiernos de Estados Unidos y la Unión Soviética. Por ello, se debió hacer un protocolo por el cual las claves secretas se entregaban físicamente a través de valija diplomática (por medio de las embajadas) para garantizar que nadie pudiera interceptar la clave electrónicamente.

2. El Reto

En este escenario tenemos a Alicia que se comunica con Roberto sobre un canal público utilizando mensajes de correo electrónico. Como a Alicia le gusta programar en Python desarrolla el algoritmo que se adjunta y acuerda con Roberto usar el criptosistema de Vernam. Queda en persona con Roberto y se intercambian varias tiras de papel ordenadas, donde cada tira tenía una lista de 10 bloques de 8 bits que sería su clave secreta para cada uno de los mensajes que se iban a enviar. Acuerdan la *codificación ASCII* para codificar sus mensajes como elementos de 8-bits con valores entre 32 al 122.

Desafortunadamente, el primer mensaje que Alicia necesitó enviar a Roberto era un párrafo muy largo, por lo que decidió hacer lo siguiente:

- dividió el párrafo en líneas de 10 caracteres
- aplicó el criptosistema de Vernam a cada línea de 10 caracteres

El código Python que utiliza para encriptar es el siguiente:

```
1 L = 10
2 for i in range(0, len(parrafo), L):
3     textos_plano.append(parrafo[i:i+L])
4     parrafo_enc = [ord(c) for c in parrafo[i:i+L]]
5     parrafo_enc = [c^k for c, k in zip(parrafo_enc, clave)]
```

```
6 textos_cifrado.append(parrafo_enc)
```

Listing 1: Implementación de fragmentación y Vernam

Alicia envía los bytes del **mensaje1** adjunto en que cada número entero es el cifrado de un bloque de 8 bits (1 byte). Cada línea es la salida de 10 caracteres. Es decir, la primera línea representa la salida de los 10 primeros símbolos, la segunda línea representa la salida cuando la entrada son los símbolos de las posiciones 10 a 20 y así sucesivamente.

Roberto decide responder, pero se encuentra con el mismo problema, su mensaje es demasiado largo por lo que decide utilizar los mismos pasos que Alicia. Siendo un fanático del «Leet Speak», utiliza un traductor¹ y además decide omitir espacios en su mensaje utilizando la segunda tira de papel con la nueva clave. Envía entonces los bytes resultantes en el **mensaje2** a Alicia.

La solución del reto tiene que detallar el contenido de los mensajes *mensaje1* y *mensaje2* en texto claro en castellano, las claves utilizadas y la estrategia seguida para recuperarlos.

¹<https://md5decrypt.net/en/Leet-translator>

Mensajes

A continuación los mensajes interceptados con su código ascii como número entero.

Mensaje 1=

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
81	231	44	130	210	66	200	155	96	188
96	190	105	190	210	83	141	141	125	178
61	178	25	137	207	85	135	145	104	253
126	253	37	137	206	69	137	141	125	184
51	226	44	158	210	17	129	145	109	180
112	243	36	137	157	64	157	150	108	179
51	253	105	157	200	88	141	145	108	174
51	225	38	130	157	93	135	140	41	175
118	225	57	131	211	66	137	157	101	184
96	178	45	137	209	17	156	141	104	169
114	255	32	137	211	69	135	223	109	184
51	246	40	152	210	66	200	154	103	253
127	243	105	137	208	65	154	154	122	188
51	235	105	143	210	92	135	223	121	168
118	246	38	204	205	94	134	154	123	176
118	178	44	130	157	82	135	145	125	188
112	230	38	204	222	94	134	223	108	177
127	253	58	194	157	98	135	140	121	184
112	250	38	204	203	80	154	150	102	174
51	241	38	129	205	80	134	154	123	178
96	178	57	153	216	85	141	145	41	184
96	230	40	158	157	71	157	147	103	184
97	243	39	136	210	17	152	158	123	169
118	178	45	137	157	93	137	223	103	178
97	255	40	152	212	71	137	223	121	184
97	253	105	157	200	88	141	141	102	253
97	247	42	131	205	88	132	158	123	253
126	243	58	204	212	95	142	144	123	176
114	241	32	131	211	17	198	223	78	175
114	241	32	141	206	17	152	144	123	253
114	246	44	128	220	95	156	158	109	178
63	178	8	128	212	82	129	158	41	—

Mensaje 2=

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
43	231	122	130	137	4	223	203	59	185
32	167	101	216	140	0	139	206	61	243
126	161	125	221	142	8	218	203	120	168
32	241	121	129	205	5	218	200	61	232
112	162	39	129	140	8	216	200	124	232
99	160	122	220	222	68	152	203	106	236
35	252	122	217	147	2	221	143	59	237
43	166	113	221	142	64	157	204	60	238
39	252	122	143	142	4	220	205	56	237
98	231	122	156	142	95	221	204	100	237
38	161	39	221	211	82	218	204	100	238
125	165	125	222	140	5	221	146	58	185
34	246	125	217	217	2	221	204	48	168
33	163	45	216	217	65	216	205	120	168
32	161	124	156	143	1	208	203	49	236
32	227	60	223	140	95	223	204	103	234
32	252	120	130	138	2	218	156	58	173
36	166	123	221	141	4	139	207	59	239
32	162	124	194	138	1	133	203	60	238
38	163	125	156	142	3	221	207	103	233
39	246	122	143	200	5	140	203	121	233
33	166	123	223	205	1	218	200	61	239
112	231	125	221	204	68	217	204	59	190
35	255	57	220	143	6	220	146	56	238
125	165	121	194	140	2	152	138	58	185
32	167	122	130	222	1	134	200	59	233
33	161	39	223	140	85	217	205	58	190
36	162	123	221	141	85	219	206	61	238
126	226	123	223	136	5	198	202	56	238
38	252	122	143	142	4	220	205	56	237
99	162	45	223	208	1	221	142	124	238
119	166	123	223	211	65	219	205	60	237
125	166	122	130	142	0	139	203	100	173
102	167	45	223	208	1	221	200	57	236
32	167	103	153	211	5	208	205	61	239
35	160	121	212	142	3	223	207	—	—