Our spies have intercepted a key exchange between Alexandria and Bobicus. We know they are using this exchange to pass a key and an initialization vector to decrypt the following ciphertext (hex form) using AES-128 in CBC mode with big-endian encoding and PKCS padding:

01c62e810475ee812688c2ef10bdd5cfe3bceb68d6ffbb2ee1d1d5d1b2653274

While we do not expect to crack AES soon, we do know they are using a crude implementation of Elliptic-Curve Diffie-Hellman to generate shared secret keys. Particularly, if $S = (x, y)$ is the shared secret, then the AES key $K = (x \ll 64) \oplus y$, and the AES initialization vector $V = (y \ll 64) \oplus x$, where $\ll$ is bitwise left-shift and $\oplus$ is bitwise XOR.

We also suspect that Alexandria's private key d_A and Bobicus' private key d_B do not refresh every message. Instead, they seem to refresh every three days, so we may have time. The original message to pass AES decryption information had these domain parameters (prefix '0x' denotes hexadecimal):

p	0xdb8f1e4884c47bfb
a	0xdb8f1e4884c47bf8
b	0xba0adf33491811a8
G	(0x18c87d6cc12ee703, 0x869a10ce9f08ed34)

With these parameters, we also captured the respective public keys involved in creating the original shared secret S:

$d_A \cdot G$	(0xc37fca91993c3e76, 0xcf780d75c662fa11)
$d_B \cdot G$	(0x1ce10c7c5989866e, 0x176acbd73cf15bc8)

Lastly, we were able to probe Alexandria's key generation service to collect two more public keys with adjusted domain parameters (see Intel.csv). Hopefully it is enough to find K and V.