import base64  
import hashlib  
import random  
from datetime import datetime  
  
class ChronoXorCipher:  
 def \_\_init\_\_(self, seed\_salt="chrono\_salt"):  
 self.seed\_salt = seed\_salt.encode()  
  
 def \_generate\_key(self, target\_time):  
 time\_str = target\_time.strftime("%Y-%m-%d %H:%M:%S")  
 seed = hashlib.sha256(time\_str.encode() + self.seed\_salt).digest()  
 random.seed(seed)  
 return bytes(random.getrandbits(8) for \_ in range(256)) # 生成256字节循环密钥  
  
 def encrypt(self, plaintext, target\_time):  
 key = self.\_generate\_key(target\_time)  
 encrypted\_bytes = bytes([p ^ key[i % len(key)] for i, p in enumerate(plaintext.encode())])  
 return base64.b64encode(encrypted\_bytes).decode() # 标准Base64编码  
  
 def decrypt(self, encrypted, target\_time):  
 key = self.\_generate\_key(target\_time)  
  
 # 修复Base64填充  
 restored = encrypted.strip()  
 missing\_padding = len(restored) % 4  
 if missing\_padding:  
 restored += '=' \* (4 - missing\_padding)  
  
 try:  
 decoded = base64.b64decode(restored)  
 return bytes([c ^ key[i % len(key)] for i, c in enumerate(decoded)]).decode()  
 except (binascii.Error, UnicodeDecodeError) as e:  
 raise ValueError("解密失败: 请检查密钥一致性或输入数据完整性") from e  
  
  
# 使用示例  
if \_\_name\_\_ == "\_\_main\_\_":  
 cipher = ChronoXorCipher()  
 target\_time = datetime(1949,10,1,15)  
  
 encrypted = 'dx3Sbk0PQ3cUrShS+p9JesjxLG+oAKUMM60S88VVO44beuOdX0EWWyjI/+G5F5kw'  
 decrypted = cipher.decrypt(encrypted, target\_time)  
 print(decrypted)

wrhklm{779ea20a94574934e72221b01bc73eee36161600}