# Lecture 5: Encryption via enciphering (part 2)

- Homework 3 will be posted later today, due Monday 2/10
- Academic conduct code: please read the syllabus + Piazza post 70
- Textbook reading for this week: *Serious Cryptography*, ch. 4, pg. 13-23

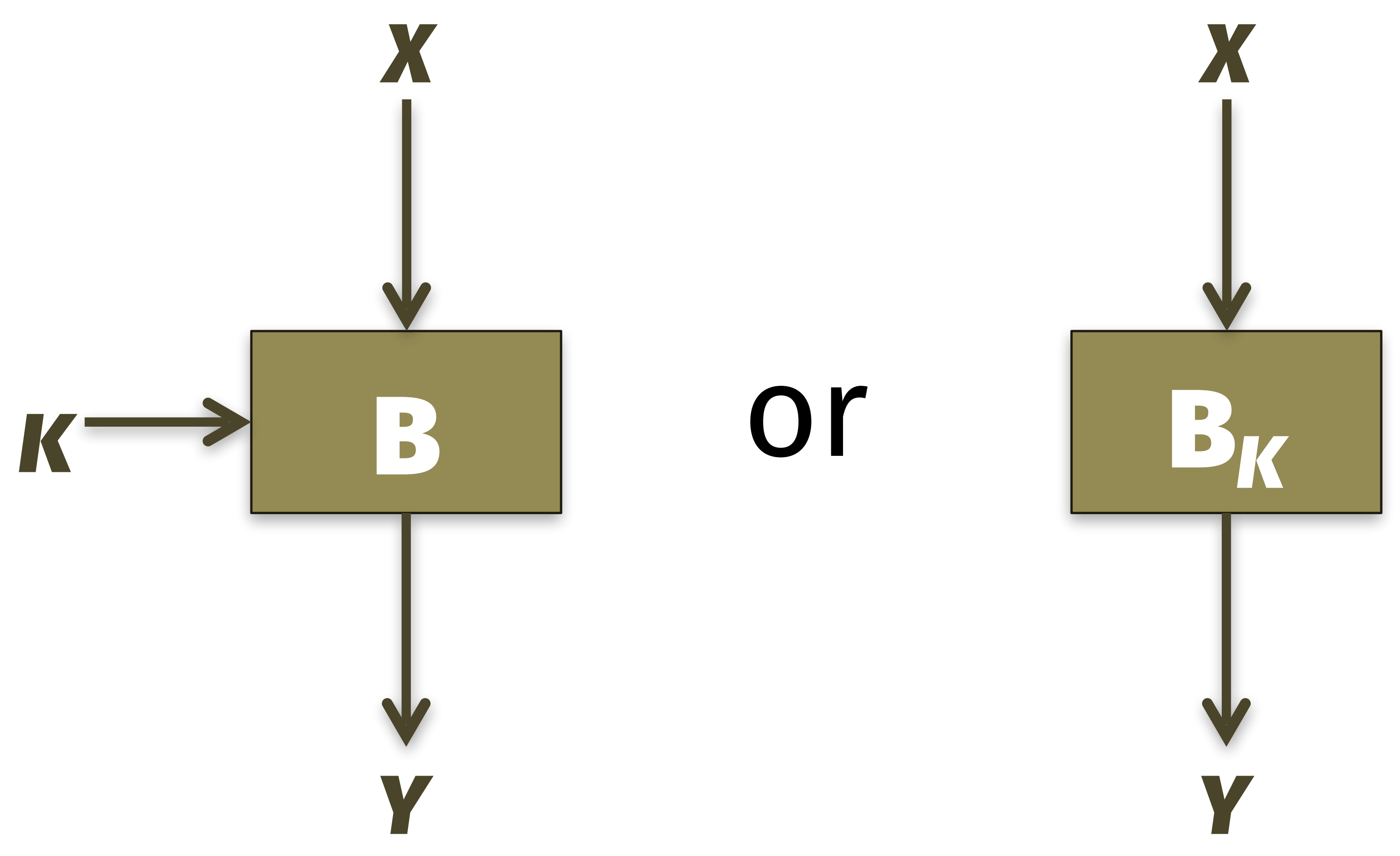
# Our plan: block ciphers → encryption

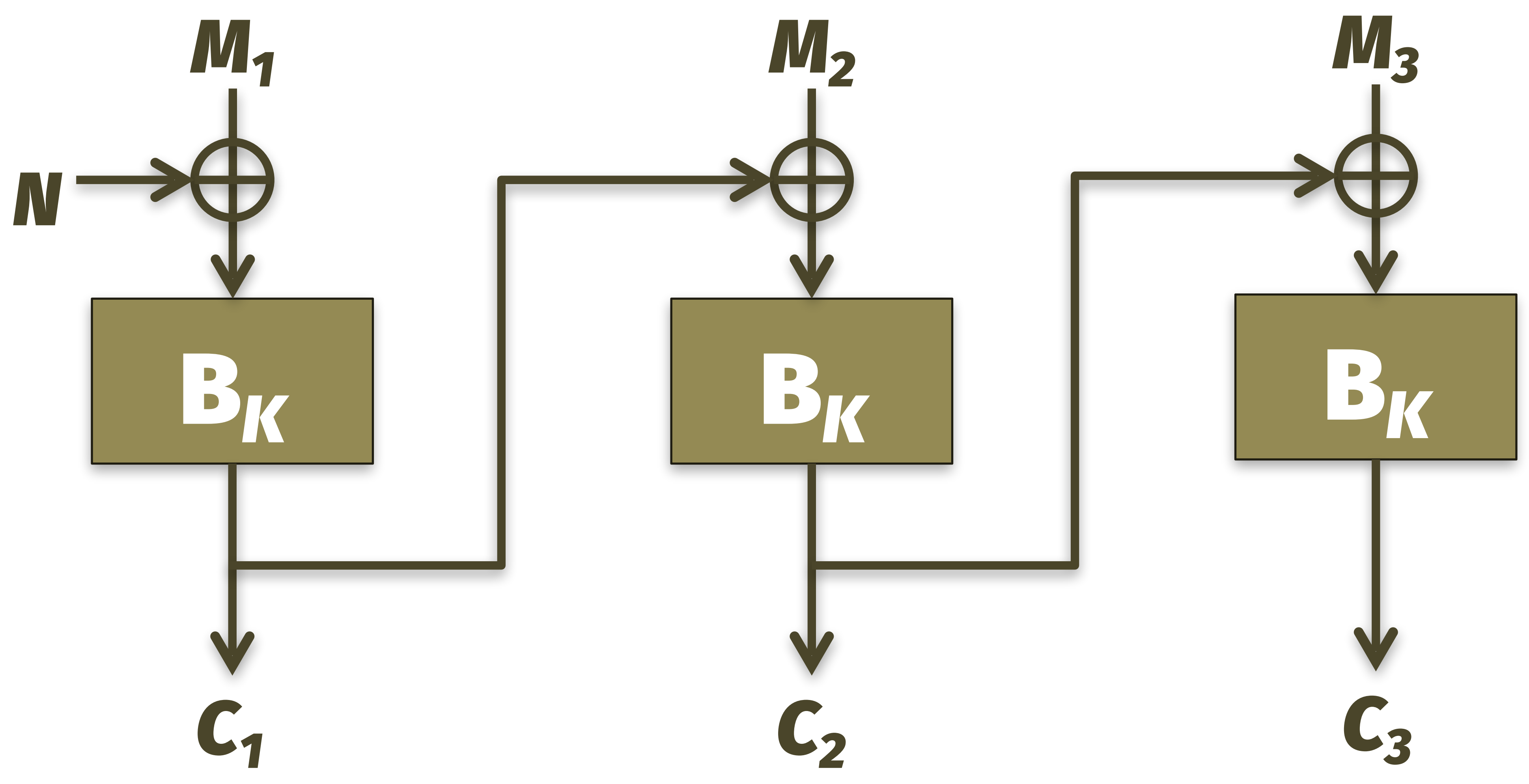
***Block cipher*** = family of codebooks

- Each key $K$ yields different codebook $B_K$
- Fast to compute: throughput ~3-4 GB/sec

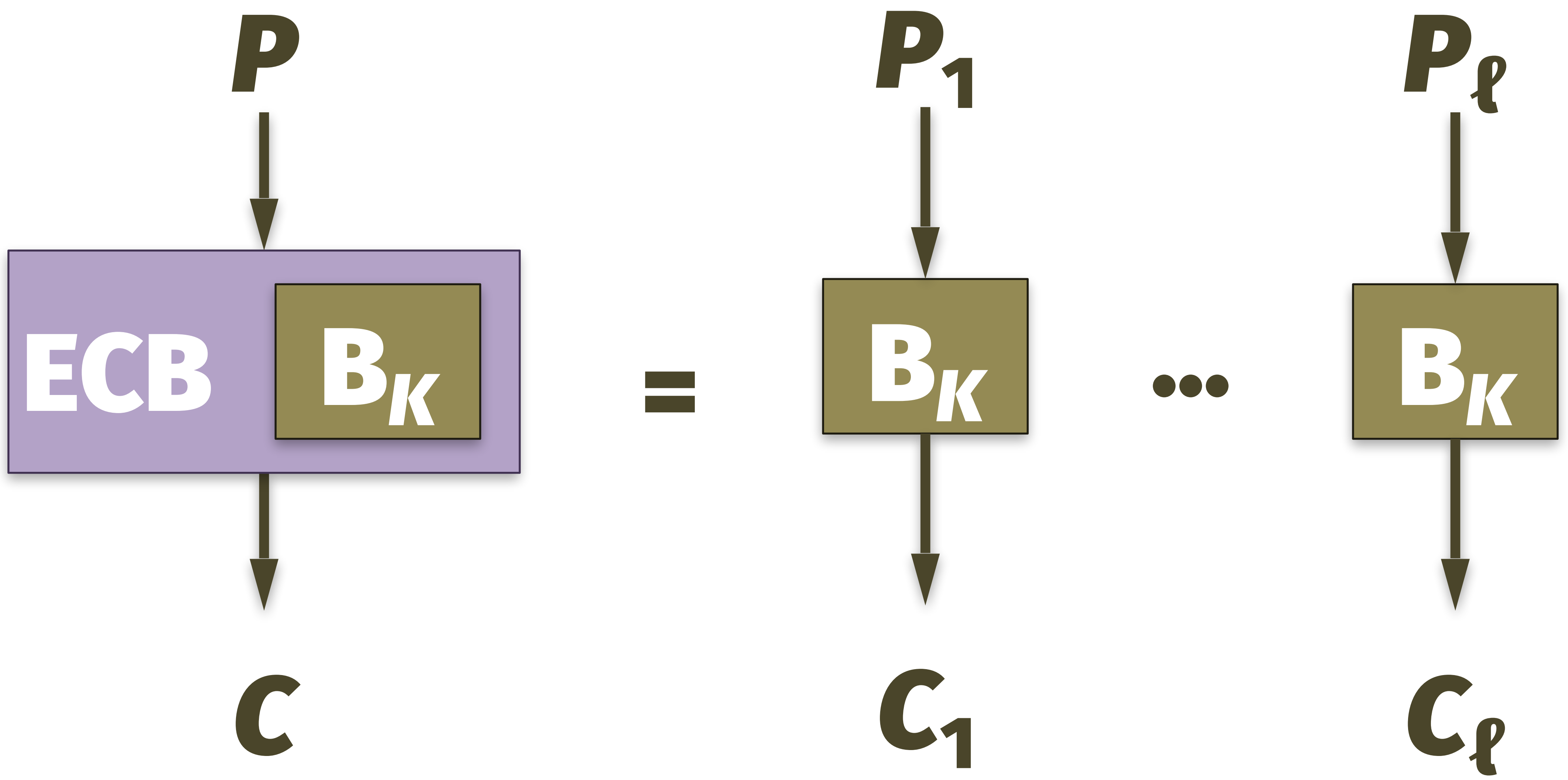
***Mode of operation*** = variability

- Allows long message with short key
- Thwarts frequency analysis



# Bad attempt: Electronic Codebook (ECB) mode




Raw image of Linux penguin

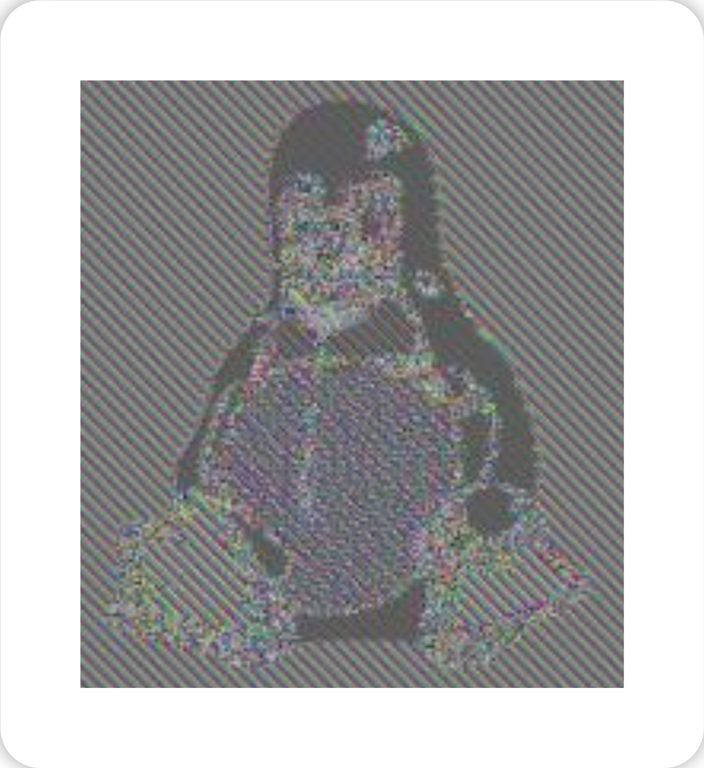
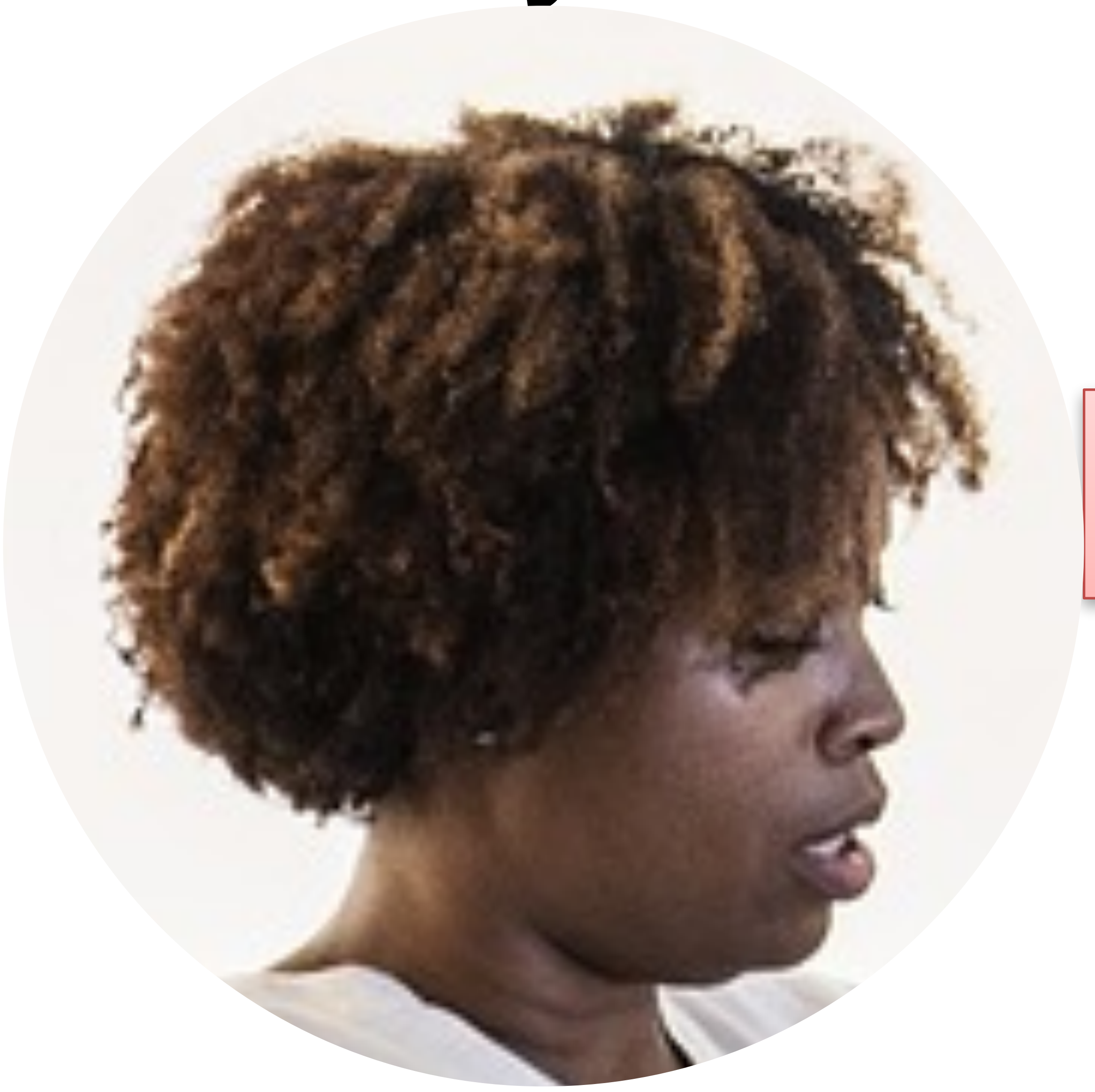


Image after ECB mode

What we want from encryption

# What if message blocks *don't* repeat?

key $K$



encode $C_i = B_K(P_i)$

key $K$

private data

$P_1, P_2, \dots P_\ell$

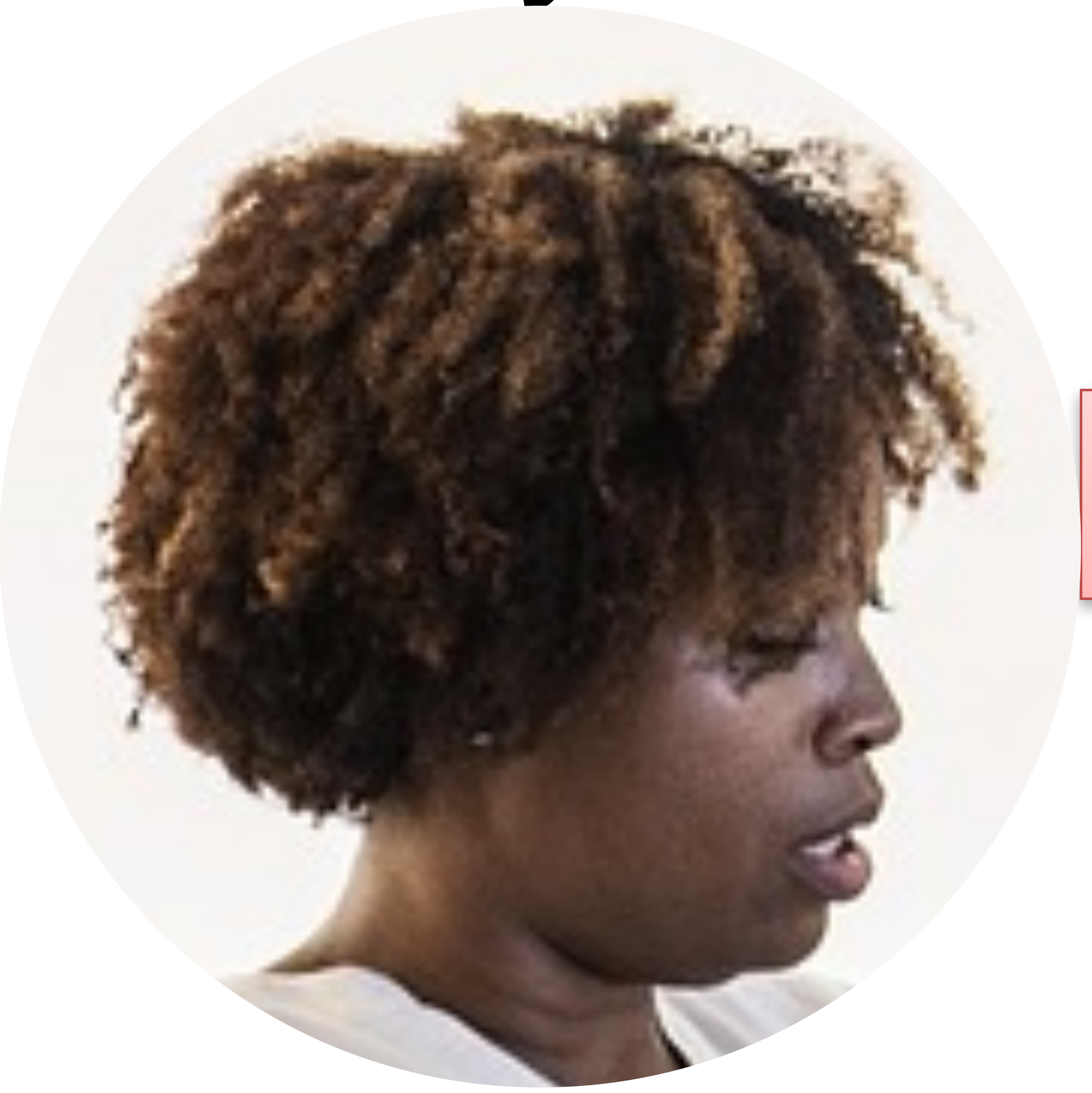
decode $P_i = B_K^{-1}(C_i)$

???

*Suppose for now that |P| is a multiple of the block length*

# What if message blocks *don't* repeat?

key $K$



private data

$P_1, P_2, \ldots P_\ell$

encode $C_i = \Pi(P_i)$

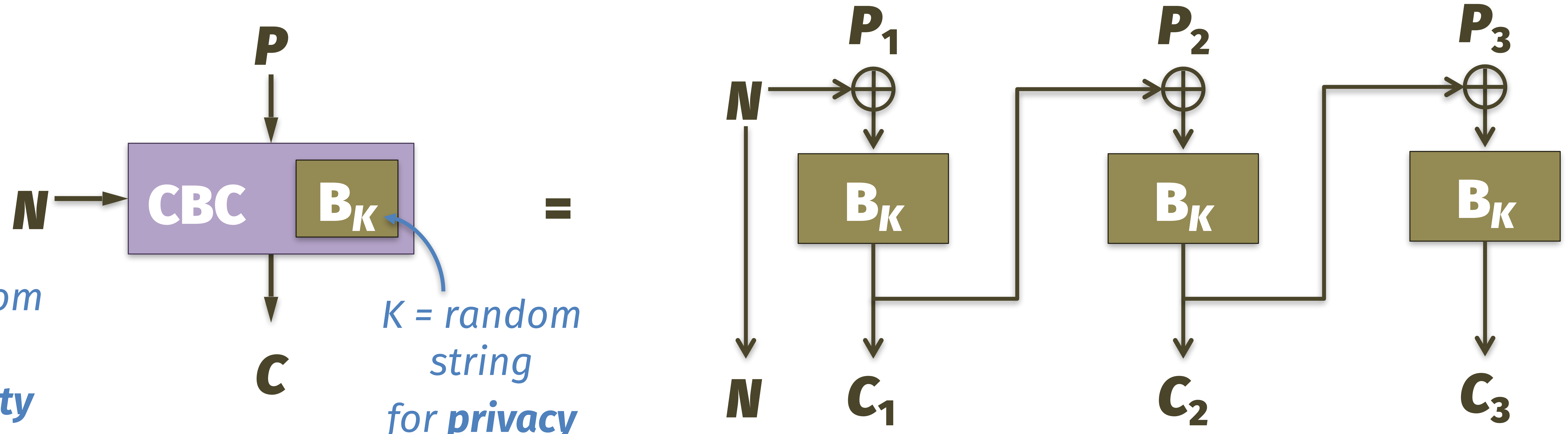
???

key $K$

decode $P_i = \Pi^{-1}(C_i)$

*How to guarantee that message blocks don't repeat?*

# Recap: CBC mode




*N = random string*

*for **variety***

*(sometimes called an initialization vector or IV)*

*K = random string*

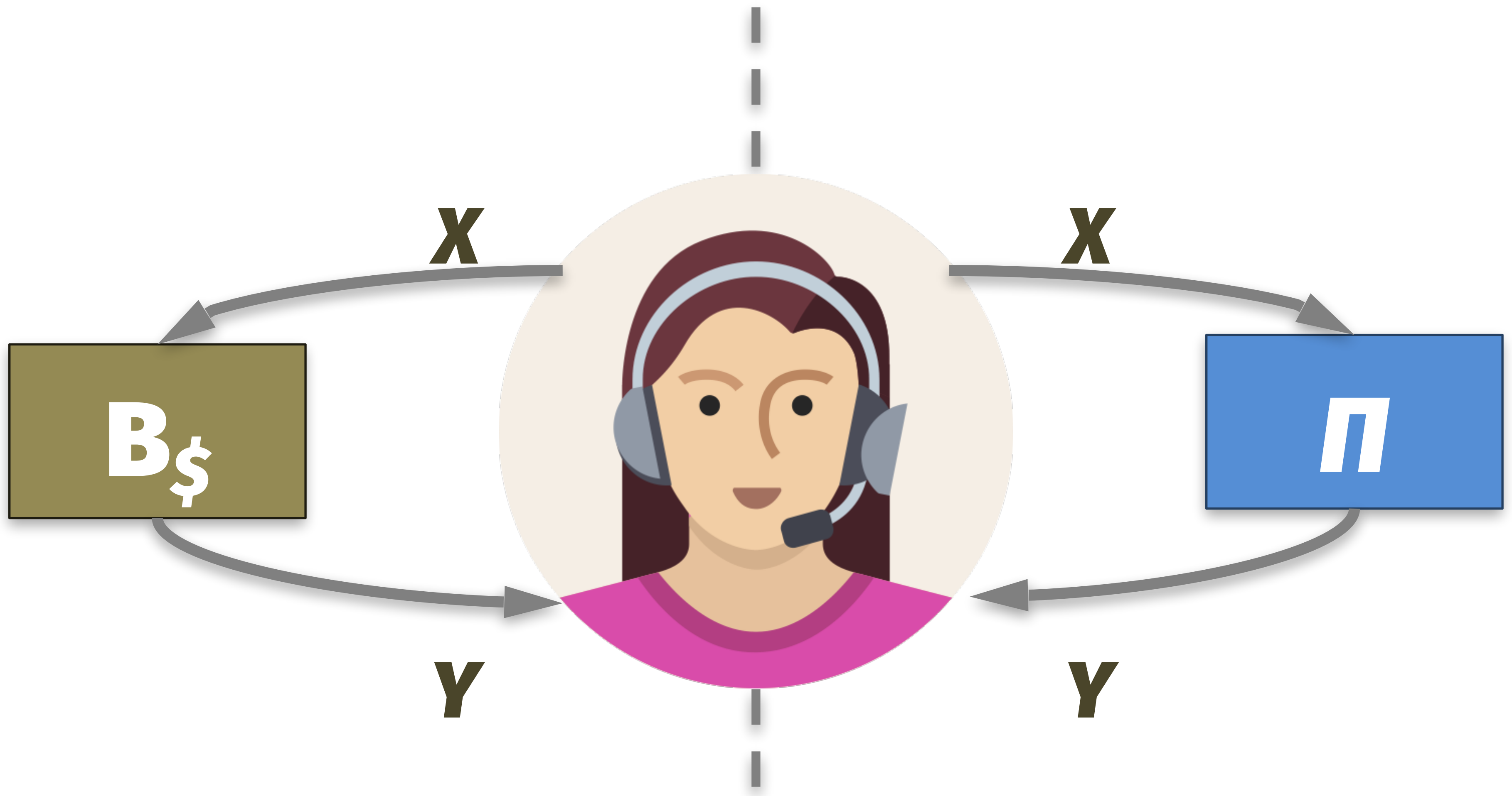
*for **privacy***

Seems like a good encryption scheme.
But how do we prove this?
In fact, what does “good" even mean?

# A new type of unpredictability

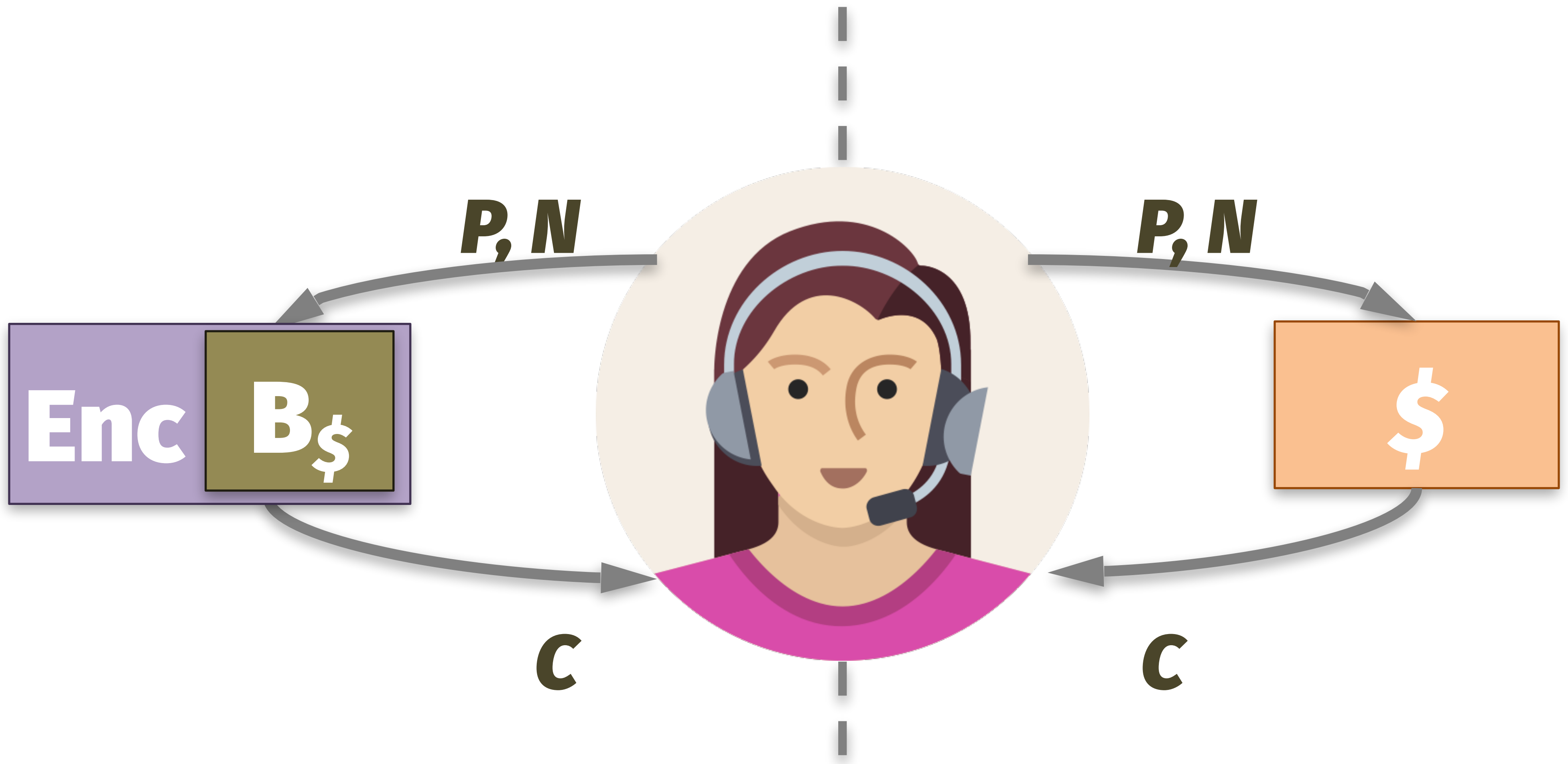
## Block cipher

$B_K$ looks like a truly random function, meaning nobody can tell them apart




## Encryption scheme

Similar, except even making the same request twice yields different answers



# Defining symmetric encryption

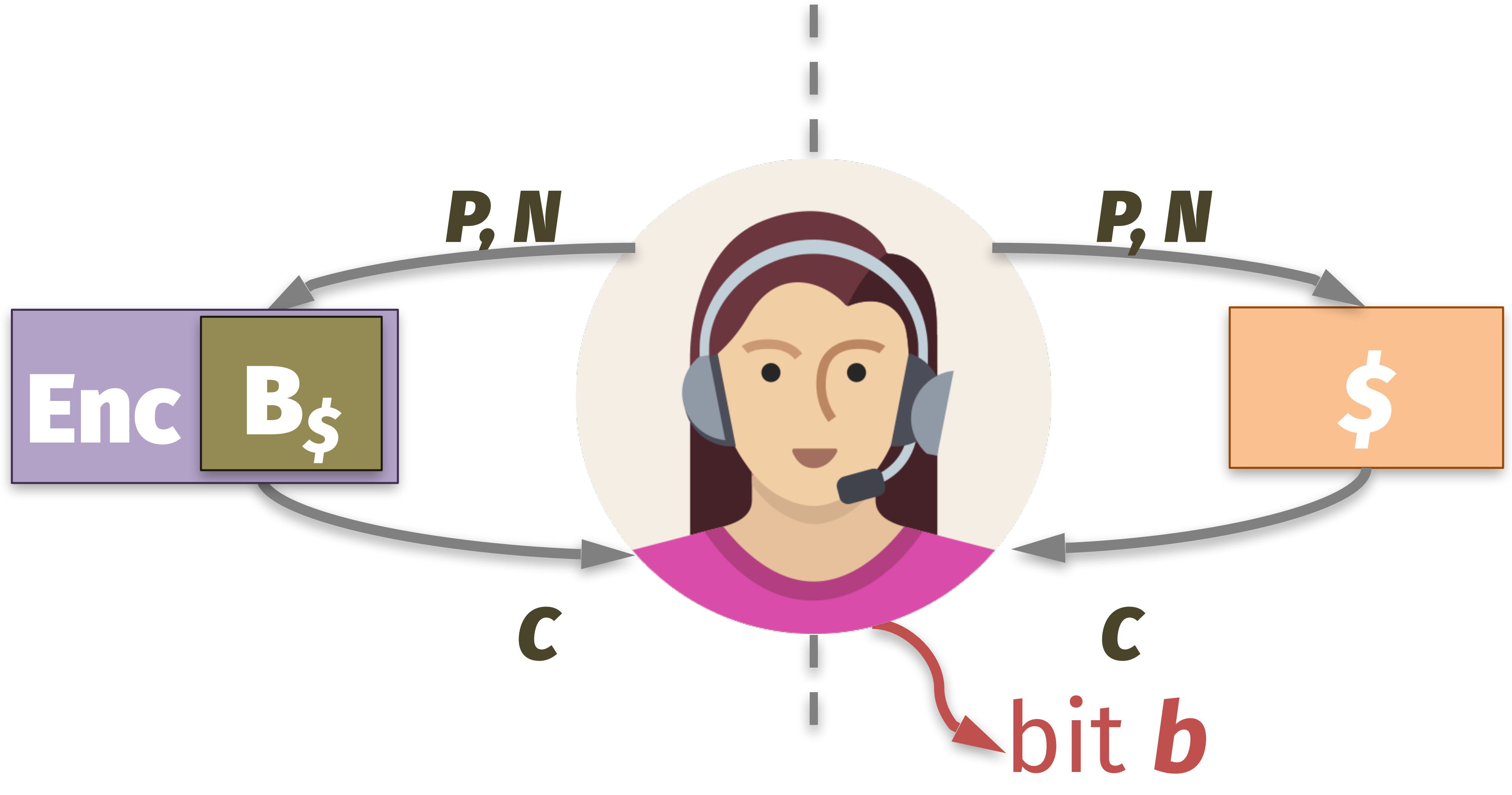
## Algorithms

- **KeyGen:** choose key $K \leftarrow \{0,1\}^{\lambda}$
- **Encrypt**$_K$$( P \in \{0,1\}^{\rho}, N) \rightarrow C \in \{0,1\}^{\gamma}$
  - Must be randomized with $\gamma \geq \rho$
- **Decrypt**$_K$$( C \in \{0,1\}^{\gamma}, N) \rightarrow P$

## Constraints

- **Performance:** All algorithms are efficiently computable
- **Correctness:** For every $K$, Enc$_K$ and Dec$_K$ are inverses
- **Security:** ???

# Pseudorandom under chosen plaintext attack (IND$-CPA)



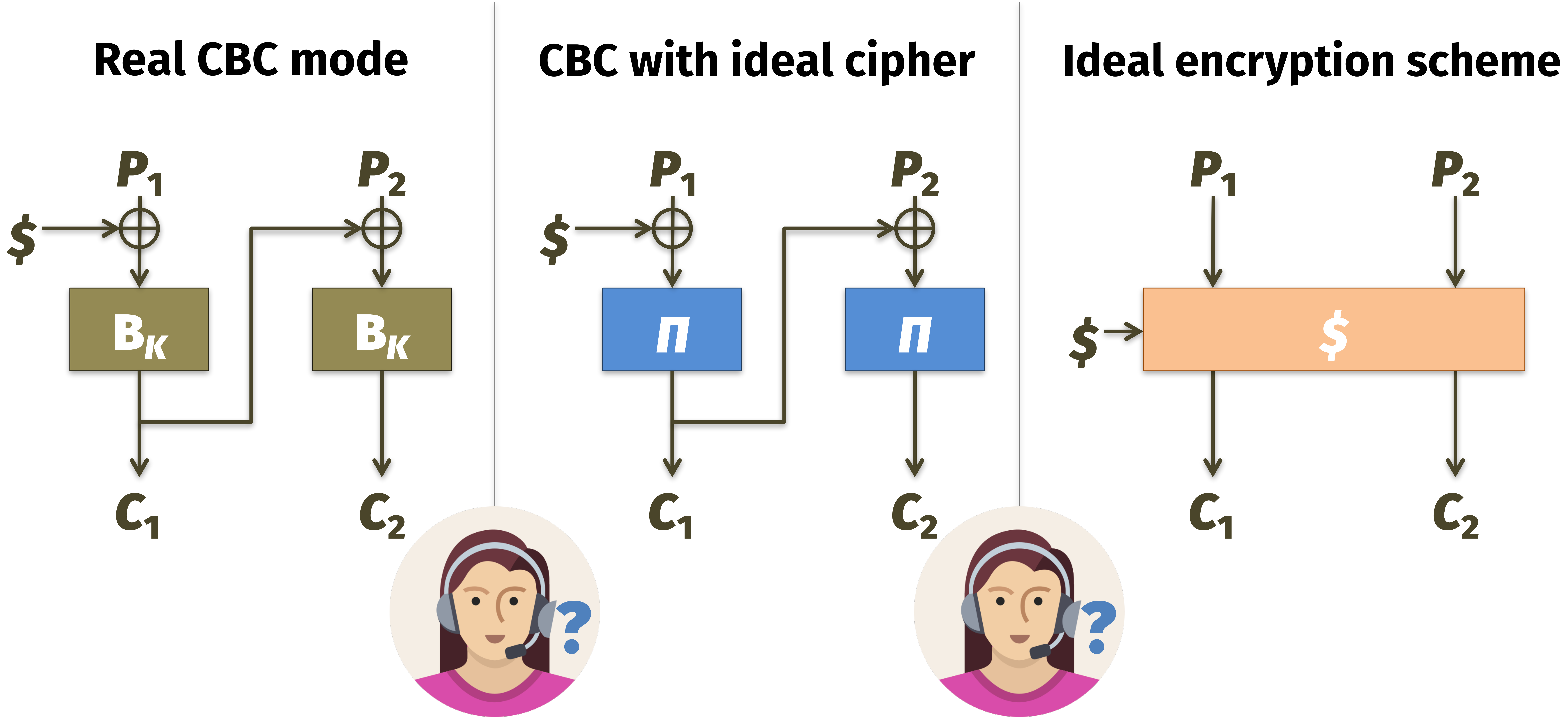

For every adv $A$ with runtime $\leq$ t and queries totaling $\leq$ q blocks,

$$A^{Enc_\$(-,-)} \approx_{(q,t,\varepsilon)} A^{\$(-,-)}$$

Two variants

- *Standard:* Eve doesn't choose *N*, instead it is chosen randomly
- *Nonce-respecting*: Eve chooses *N*, but each choice must be distinct

# Informal proof by picture: CBC mode is IND$-CPA




*by pseudorandomness of the block cipher*

*input to every Π is "random"*

# Encryption in practice

## bu.edu homepage (2017)

**Obsolete connection settings**

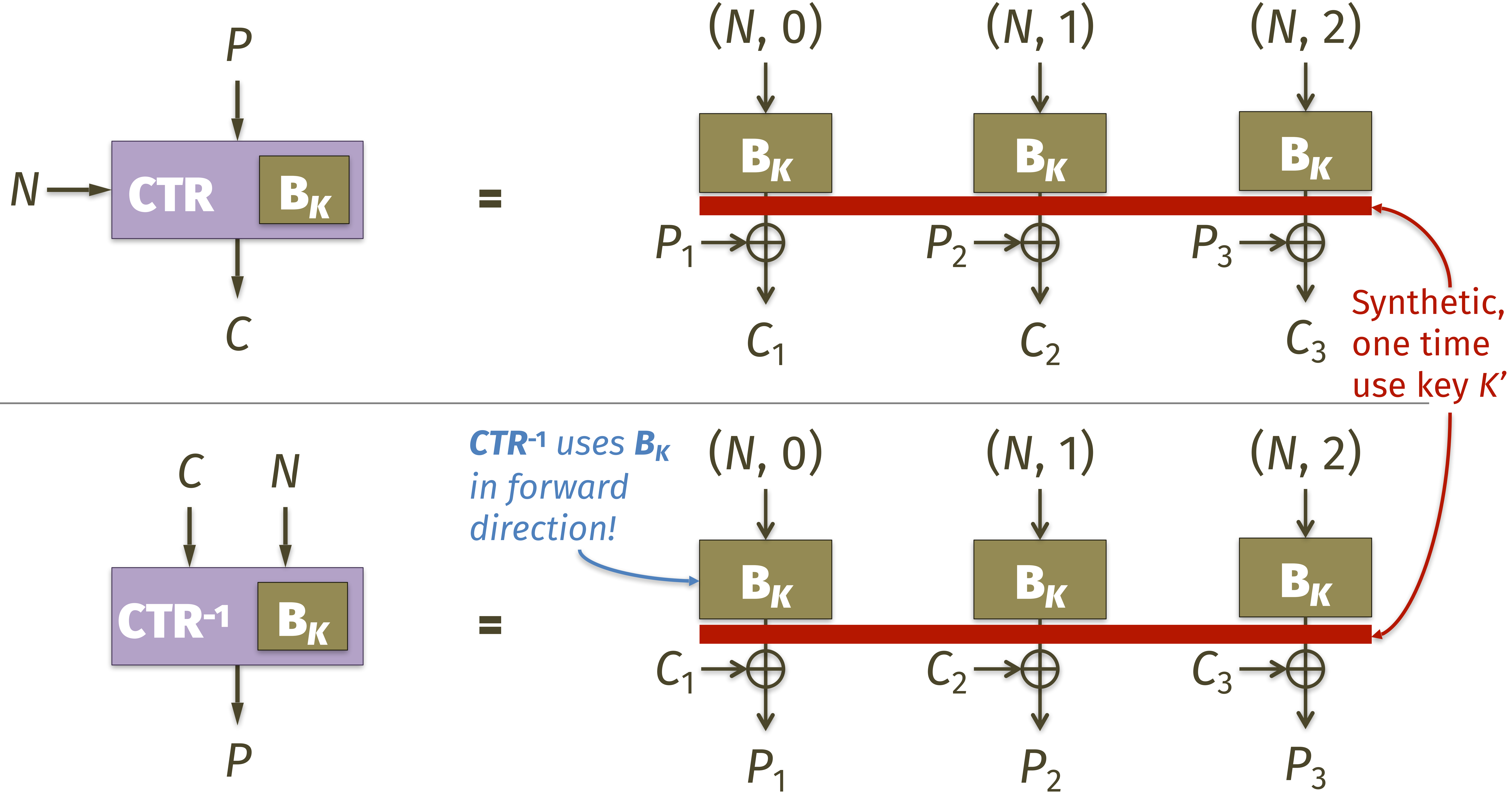
The connection to this site uses TLS 1.0 (an obsolete protocol), RSA (an obsolete key exchange), and AES_256_CBC with HMAC-SHA1 (an obsolete cipher).

## www.amazon.com

**Secure connection**

The connection to this site is encrypted and authenticated using TLS 1.2 (a strong protocol), ECDHE_RSA with P-256 (a strong key exchange), and AES_128_GCM (a strong cipher).

# Counter (CTR) mode



# Issues to consider with CTR mode

96 bits 32 bits

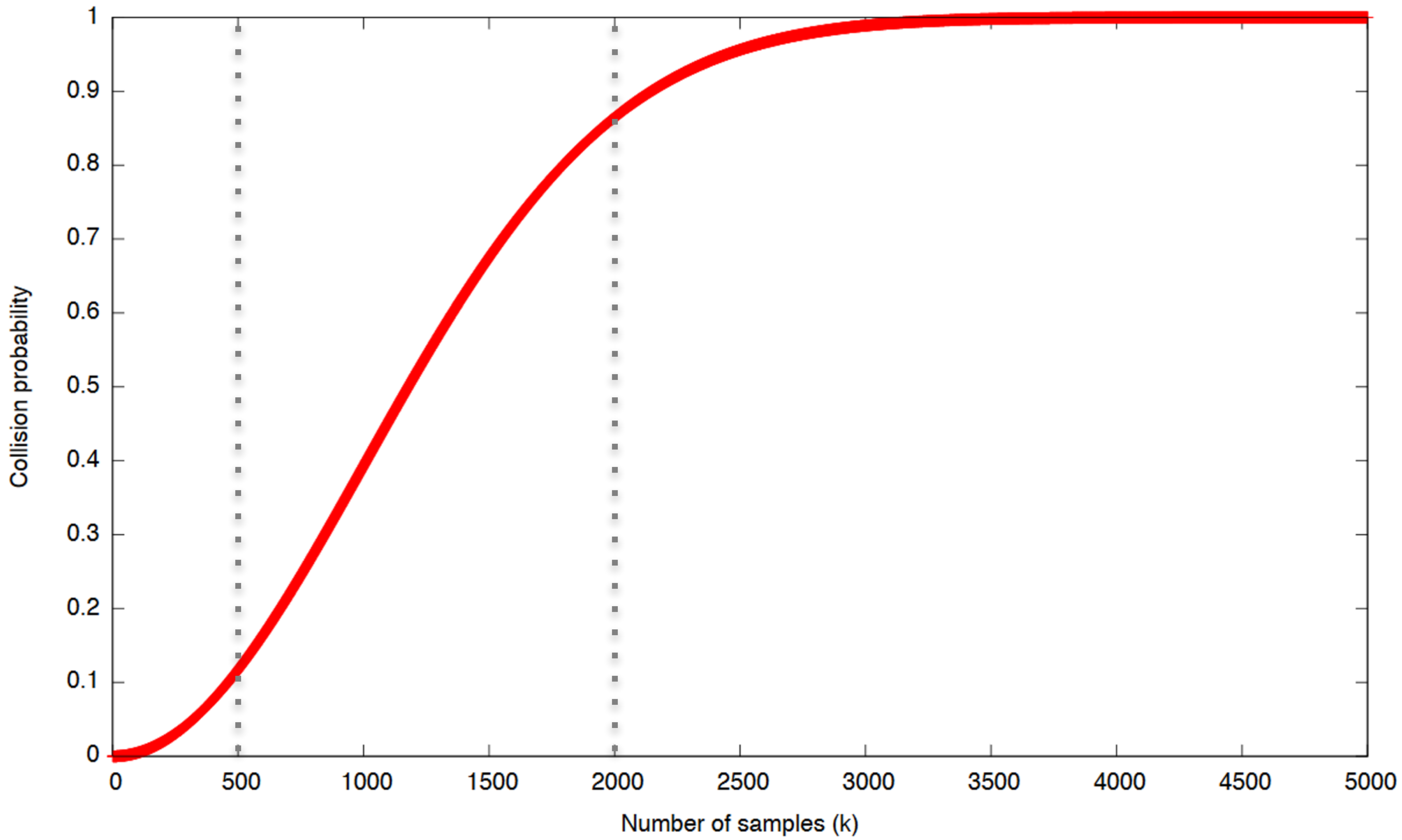
1. Tradeoff between the lengths of ***N*** and ***ctr***

2. How do we choose ***N*** if the parties are stateless? → *choose randomly, rely on birthday bound*

3. How to prove that CTR satisfies IND$-CPA? → *later today*

4. What to do if ***N*** is accidentally repeated? → *will re-visit in a few weeks*

# Birthday bound

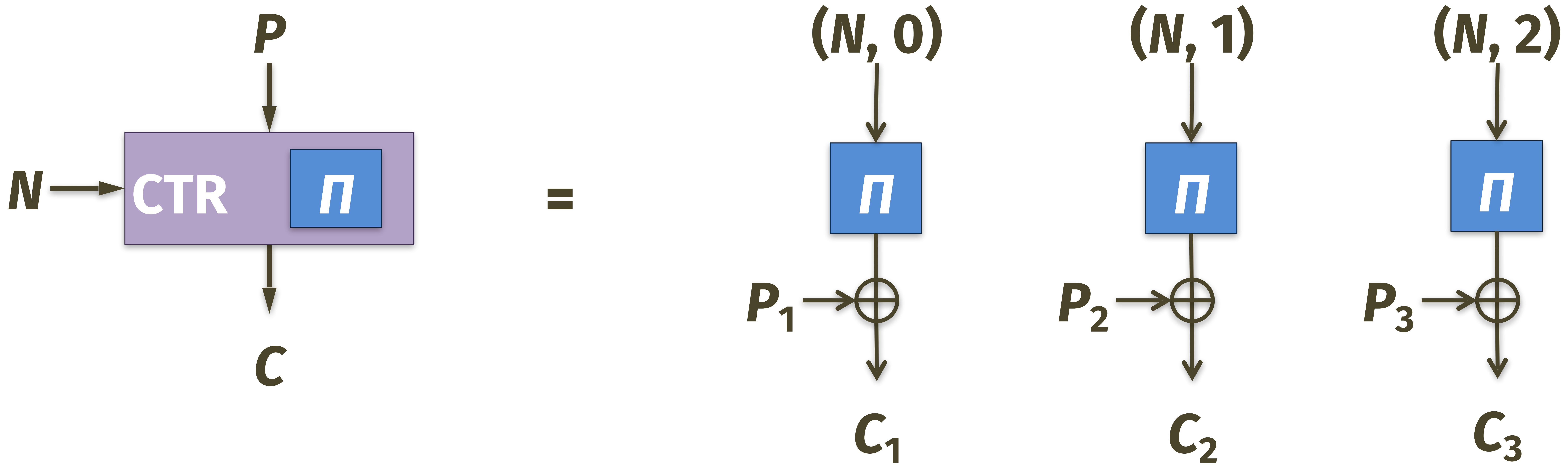



- When drawing with replacement from set of size L,

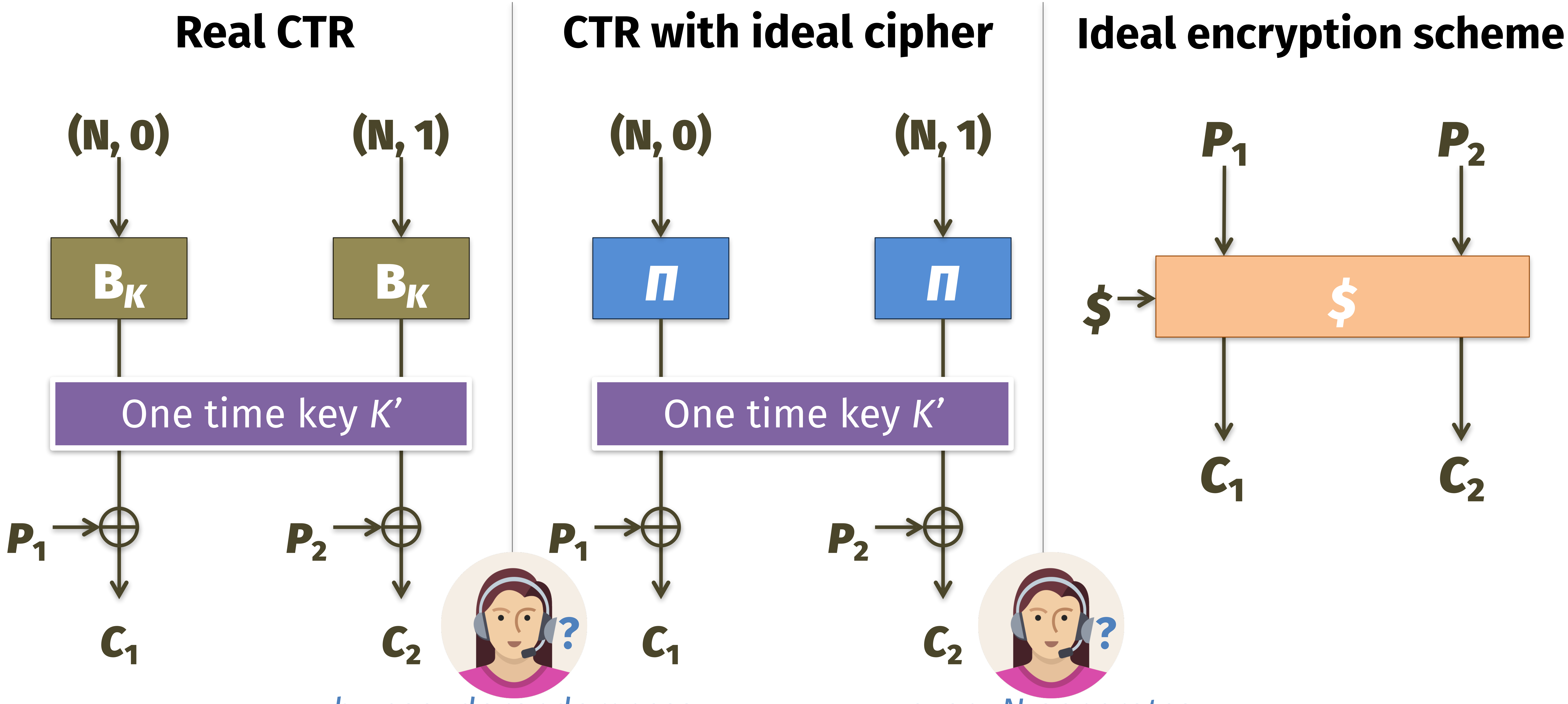
$$E[\text{\# items to draw until first collision}] \approx \sqrt{\frac{\pi}{2}L} \approx 1.25\sqrt{L}$$

- The distribution of M is tightly concentrated around its expected value

# Observation: CTR mode with $\Pi \Rightarrow$ one time pad



# Informal proof that CTR mode is IND$-CPA




*by pseudorandomness of the block cipher*

*every N generates a unique key*

# Toward a mathematically rigorous proof

<u>If we begin with:</u>

a block cipher $B_K$
that is ($q_B$, $t_B$, $\varepsilon_B$) pseudorandom

<u>Then we can construct:</u>

symmetric key enc scheme **Mode** $B_K$
that is ($q_C$, $t_C$, $\varepsilon_C$) indistinguishable from pseudorandom under a chosen plaintext attack

***Contrapositive*:** If an adversary ***A*** can solve the IND$-CPA game on **Mode** $B_K$, then we can construct an adversary ***A'*** that breaks $B_K$ almost as effectively.

# Thm: $B_K$ is pseudorandom → CTR $B_K$ is IND\$-CPA

<u>If we begin with:</u>

Adversary $\boldsymbol{A}_{\mathbf{CTR}}$ who can distinguish

CTR $B_K$ from \$

with probability > $\varepsilon_C$ given time $t_C$ and queries that total $q_C$ blocks of data

<u>Then we can construct:</u>

Adversary $\boldsymbol{A}_{\mathbf{BC}}$ who can distinguish

$B_K$ from $\Pi$

with probability > $\varepsilon_B$ given time $t_B$ and a total of $q_B$ queries

# Formal CTR mode reduction

If we begin with:

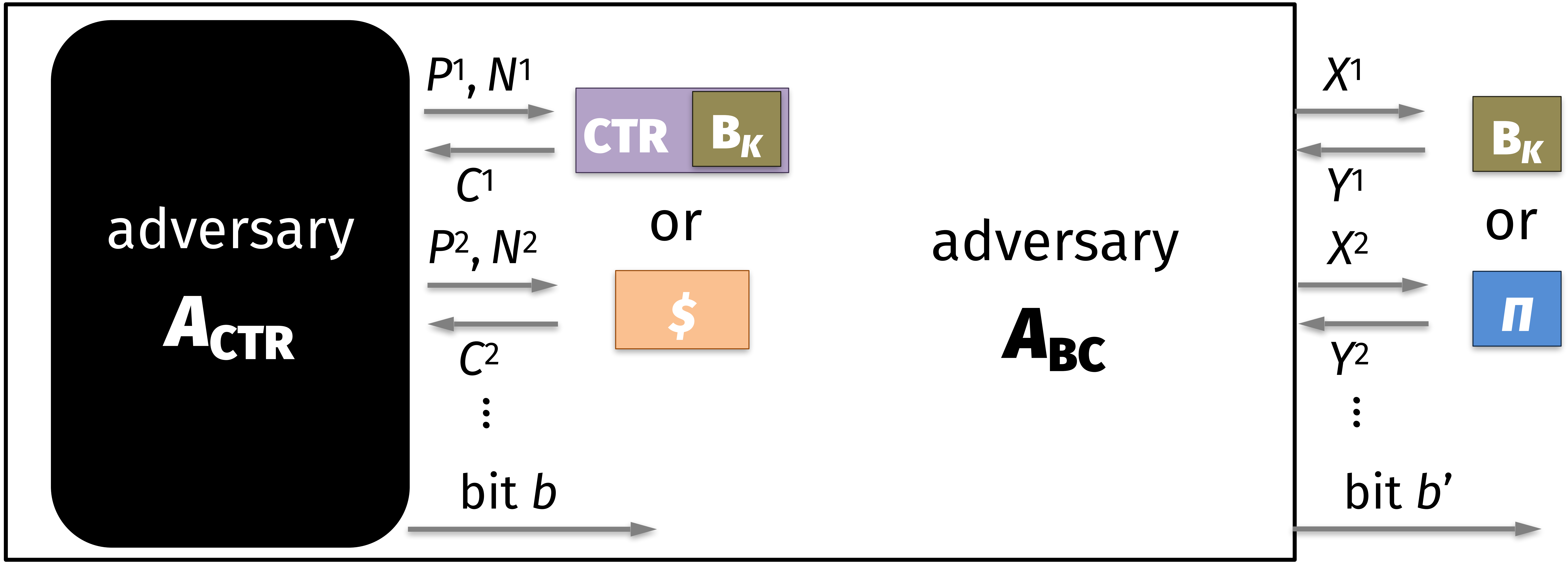
Adversary $A_{CTR}$ who can distinguish

CTR $B_K$ from \$

Then we can construct:

Adversary $A_{BC}$ who can distinguish

$B_K$ from $\Pi$



# How $A_{BC}$ operates

*Step 1:*

Wait for $A_{CTR}$ to output a ($P$, $N$) pair

*Step 2:*

Query $A_{BC}$'s own oracle on (N,0), (N,1), ..., (N, |$P$|-1)

*Step 3:*

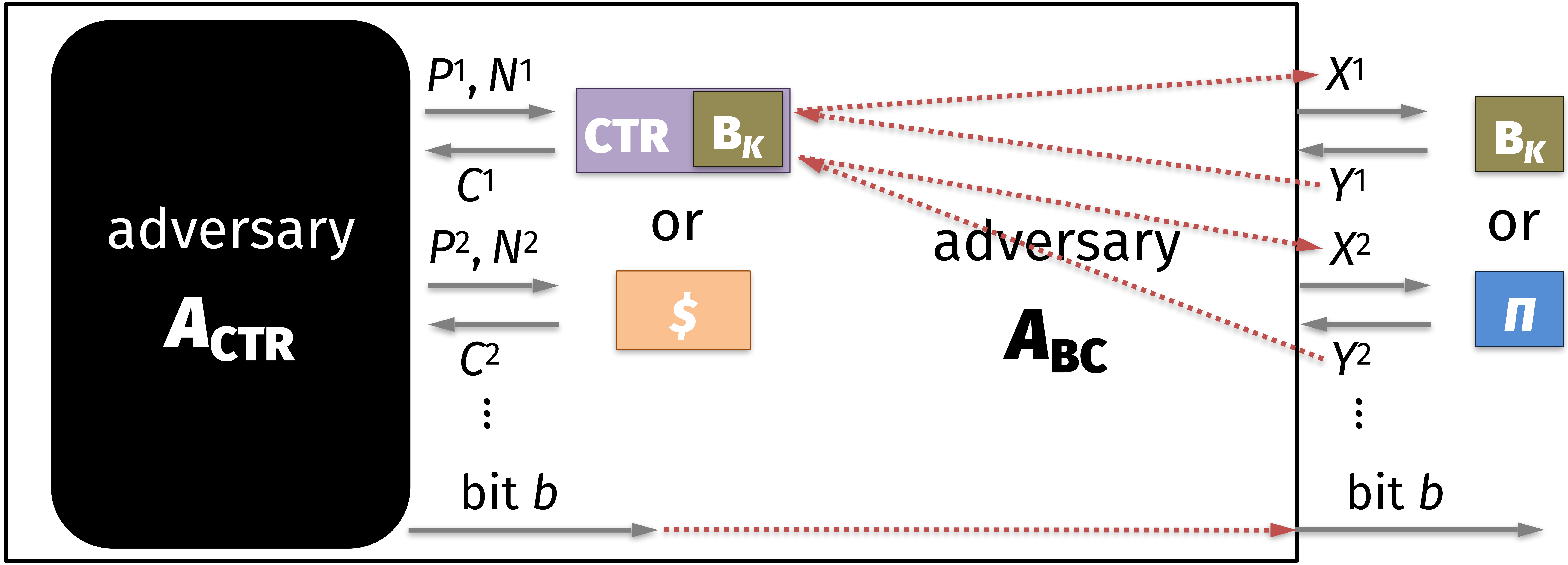
Concatenate response blocks, then xor with $P$

*Step 4:*

Repeat

*Step 5:*

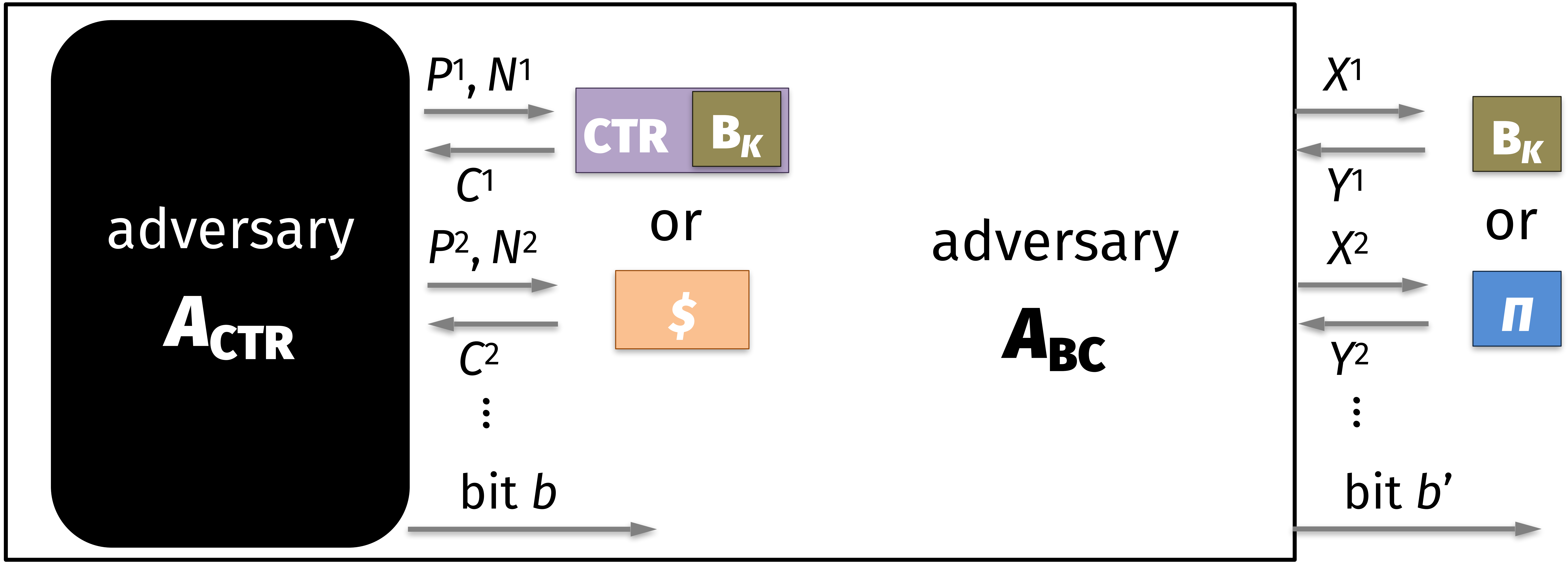
Output the same bit as $A_{CTR}$



# Why this reduction works

If $A_{BC}$ is talking to $B_K$, then this procedure faithfully yields CTR $B_K$

If $A_{BC}$ is talking to $\Pi$, then this procedure faithfully yields CTR $\Pi$, identical to $ by non-repetition



# Our final result

<u>If we begin with:</u>

Adversary $\boldsymbol{A}_{\mathbf{CTR}}$ who can distinguish

CTR $B_K$ from \$

with probability > $\varepsilon_C$ given time $t_C$ and queries that total $q_C$ blocks of data

<u>Then we can construct:</u>

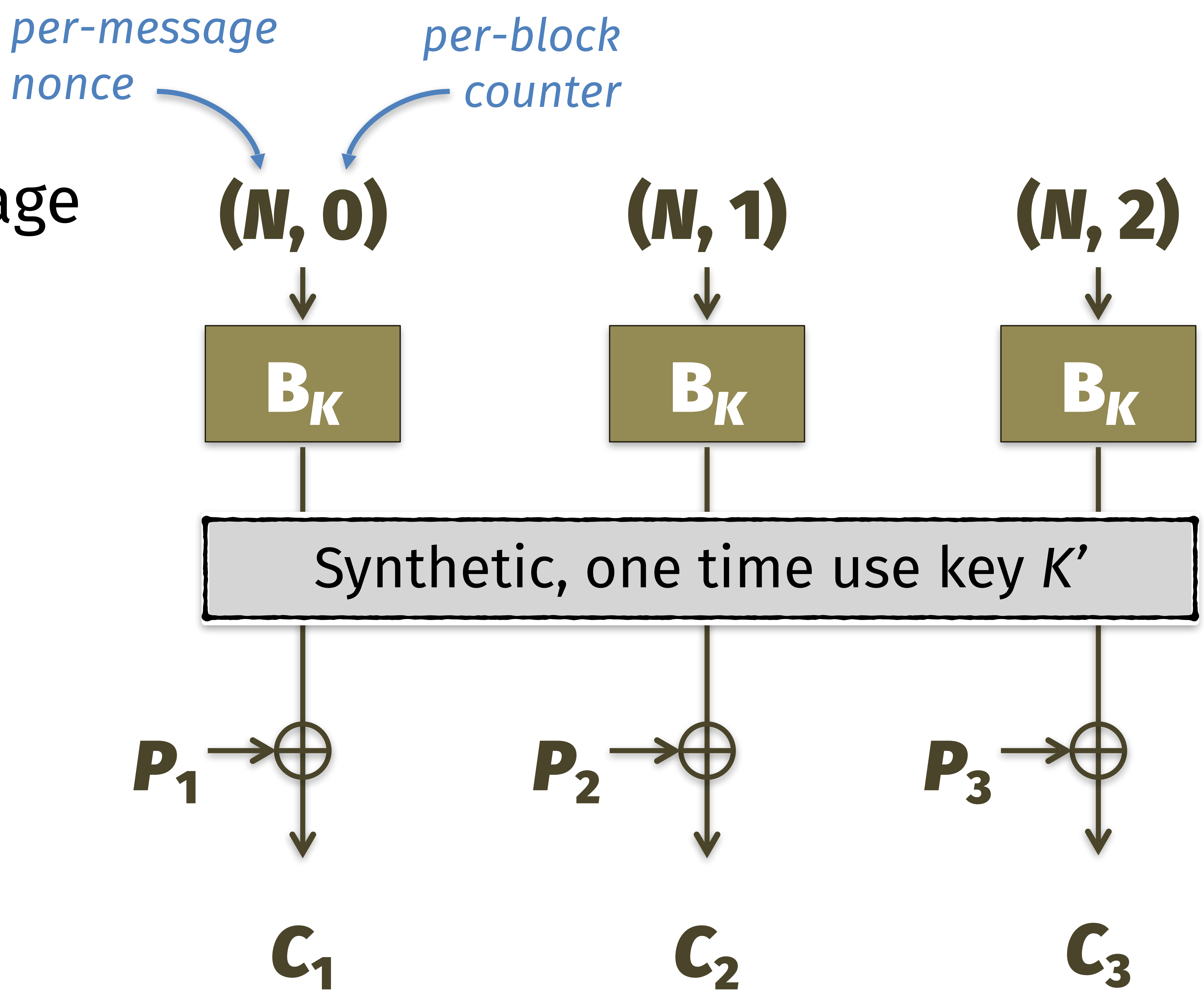
Adversary $\boldsymbol{A}_{\mathbf{BC}}$ who can distinguish

$B_K$ from $\Pi$

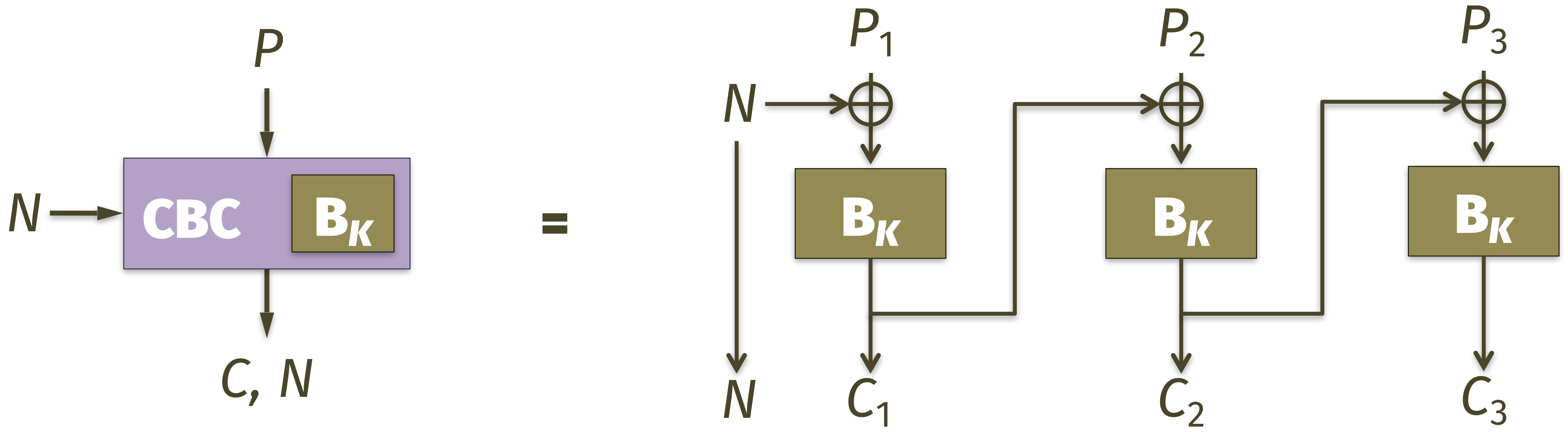
with probability > $\varepsilon_C$ given time $t_C + q_C$ and a total of $q_C$ queries

# What if $|P|$ isn't a multiple of the block length?

- CTR mode produces a keystream to XOR with message
- If you don't need the full keystream, just discard it
- No need to pad in CTR



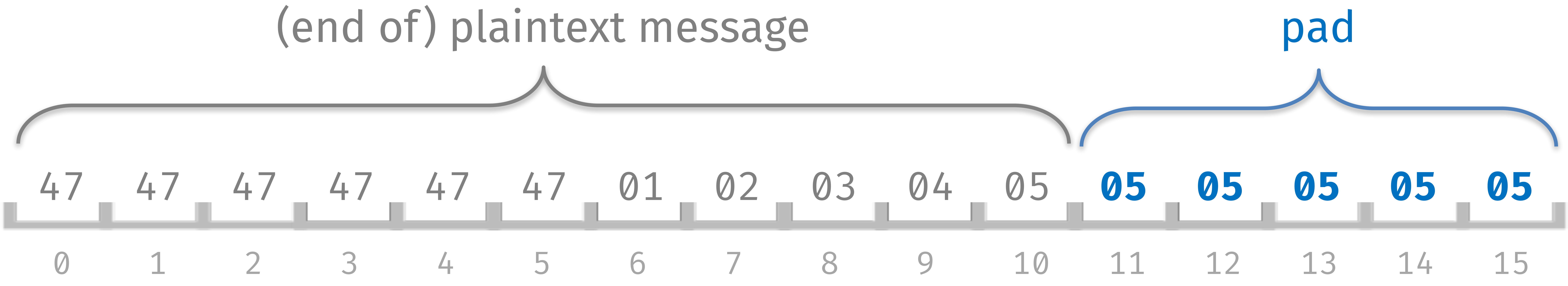
# Padding in CBC?




- Not as simple: $B_K$ requires exactly 1 block of text, which means the XOR needs two inputs that are 1 block long
- Seems like padding the final block $P_3$ is necessary…

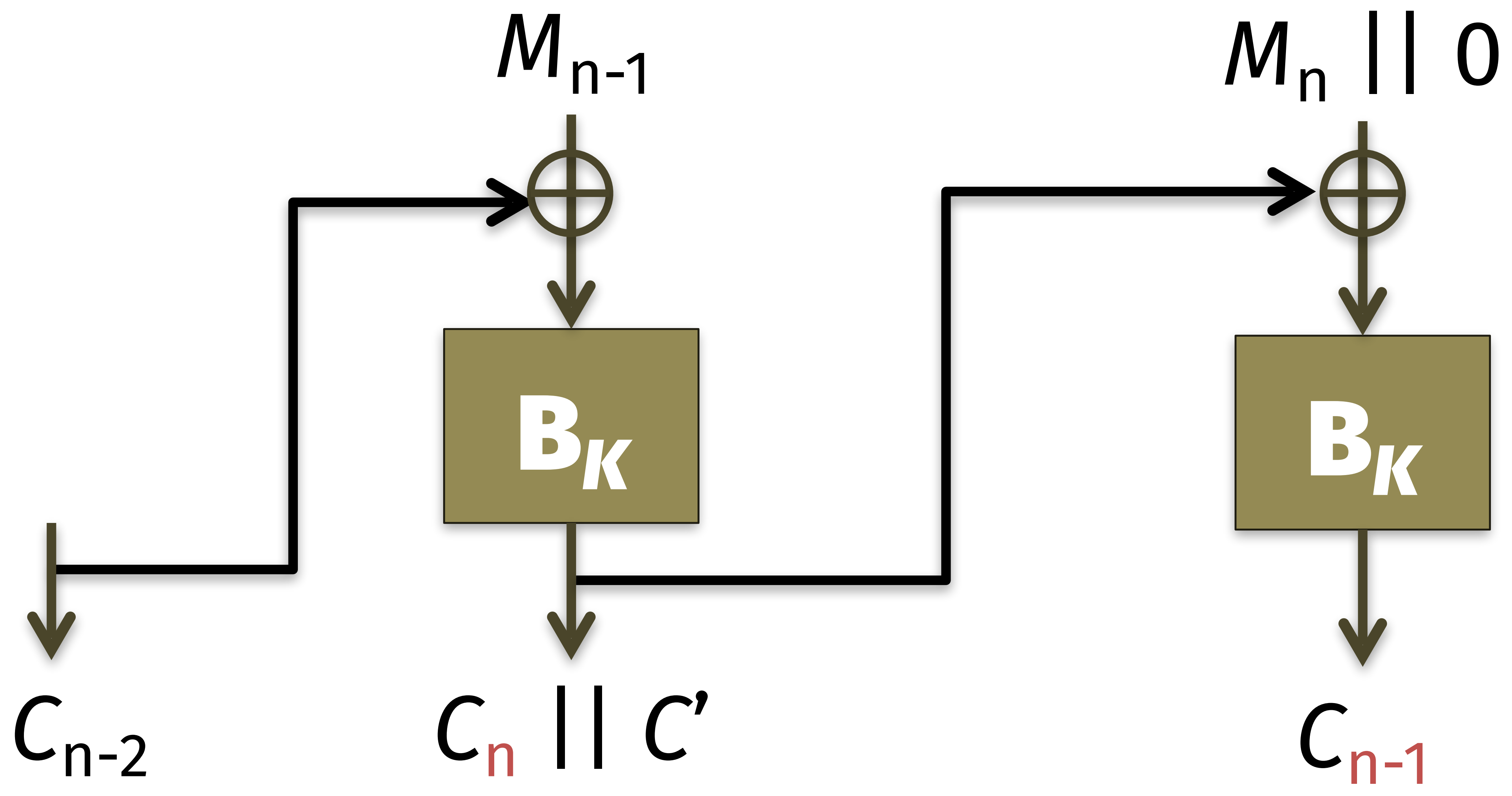
# PKCS #7 padding

Padding adds M whole bytes, each of value M




What if the message is already a multiple of the block length?

# Ciphertext stealing for CBC




<u>How to encrypt</u>

- Pad the final block with 0s
  (on its own, this is not invertible)
- Output the entire final block
- For the second-to-last block,
  only output the first $|M_n|$ bytes

# Ciphertext stealing for CBC

<u>How to encrypt</u>

- Pad the final block with 0s
  (on its own, this is not invertible)
- Output the entire final block
- For the second-to-last block, only output the first $|M_n|$ bytes

<u>How to decrypt</u>

- First decrypt the last block
- Data after the first $|M_n|$ bytes == $C'$
- Now can decrypt the penultimate block