# Assignment #8 – Solutions

## Problem 1

Each participant $T_i$ selects a random polynomial

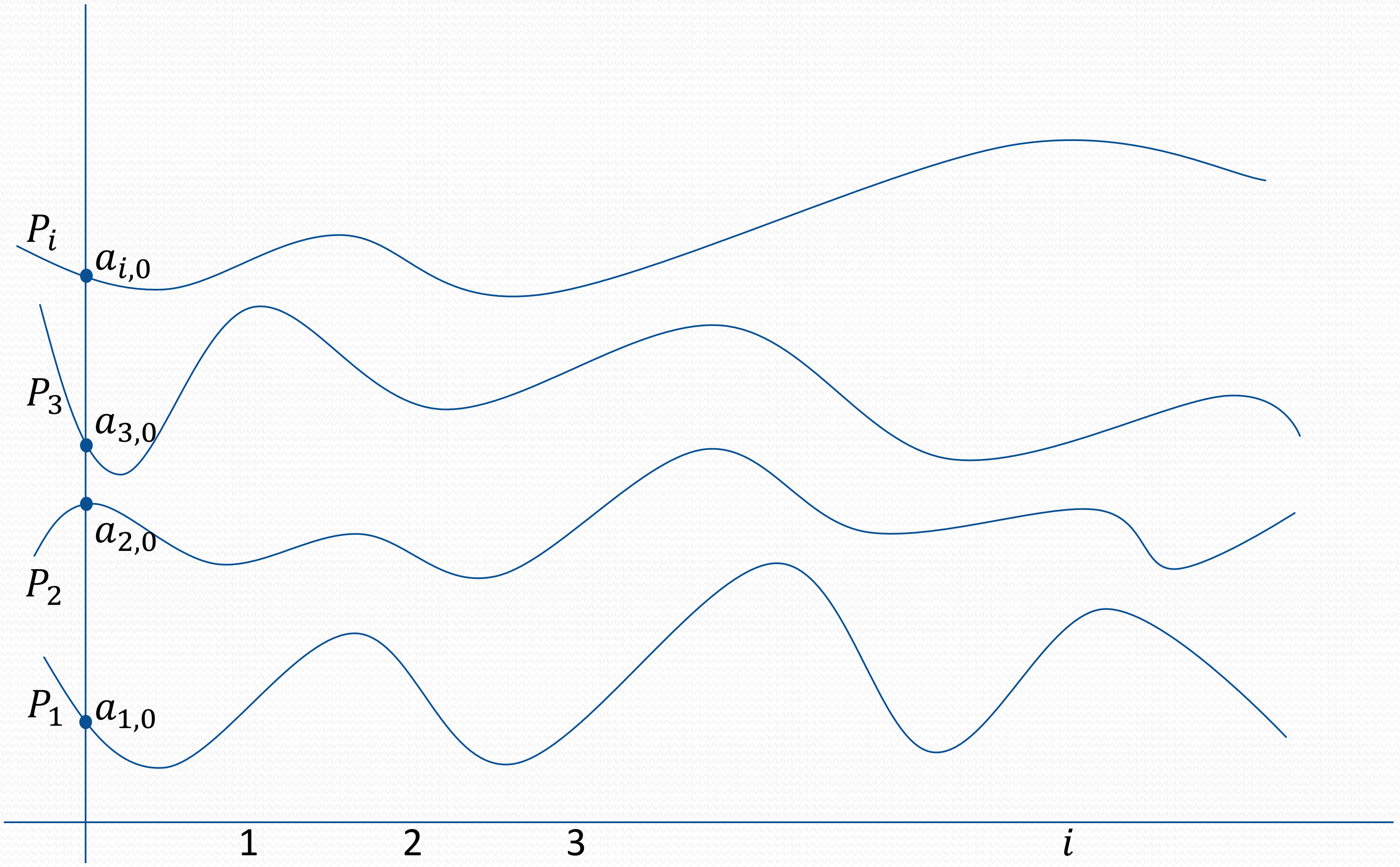
$$P_i(x) = \sum_{j=0}^{k-1} a_{i,j} x^j \bmod q$$

The joint secret is the sum of the original secrets

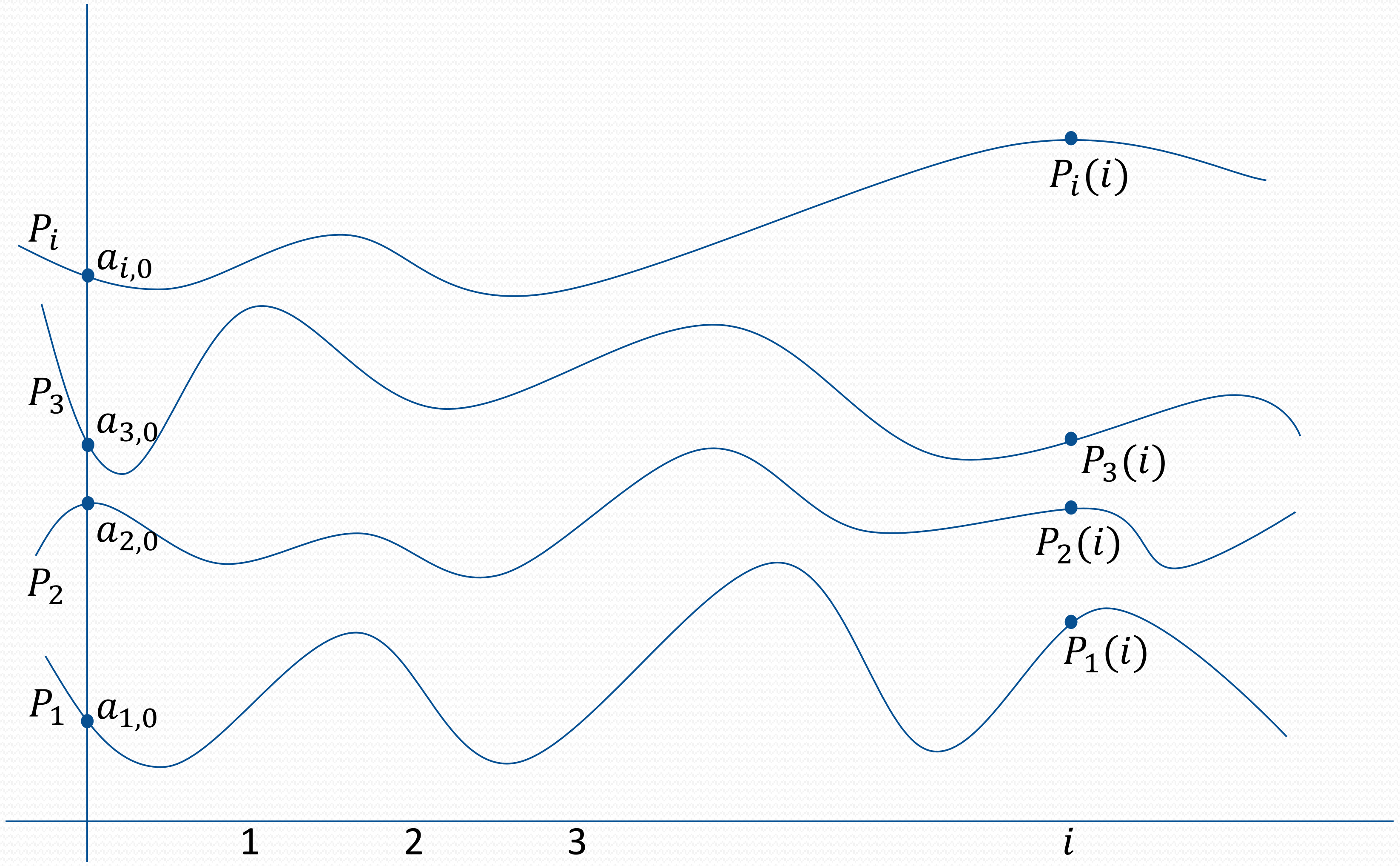
$$\sum_{i=1}^{n} P_i(0) \bmod q = \sum_{i=1}^{n} a_{i,0} \bmod q$$

How are shares of the joint secret formed?

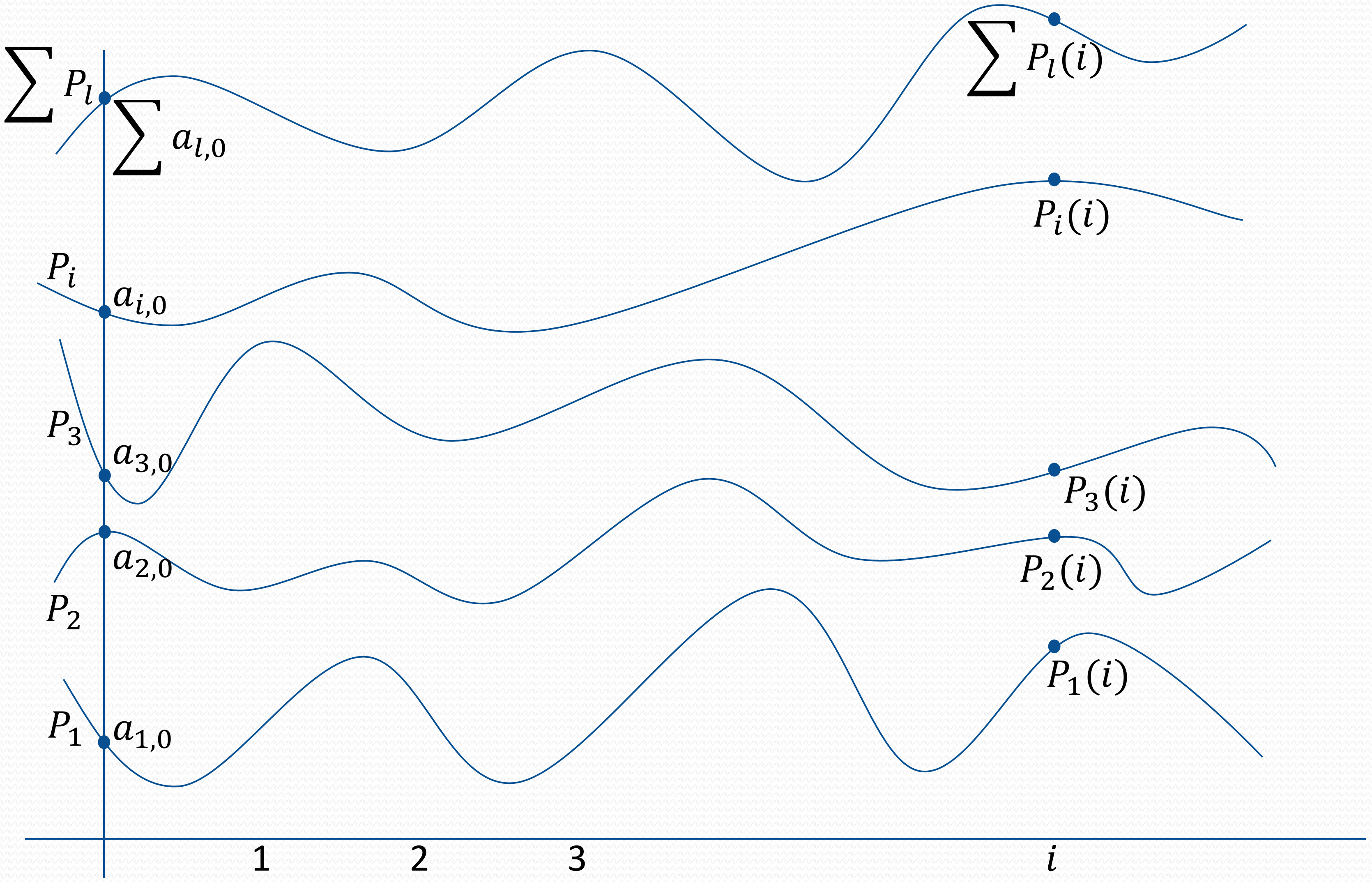
# Problem 1



# Problem 1



# Problem 1


$\sum P_l$
$\sum a_{l,0}$
$\sum P_l(i)$
$P_i(i)$
$P_i$
$a_{i,0}$
$P_3$
$a_{3,0}$
$P_3(i)$
$a_{2,0}$
$P_2(i)$
$P_2$
$P_1(i)$
$P_1$
$a_{1,0}$
1
2
3
$i$

# Problem 1

Participant $T_i$ computes its share of the secret $S = \sum_{l=1}^{n} a_{l,0} \bmod q$ by taking the values $P_1(i), P_2(i), \ldots, P_n(i)$ (including the value $P_i(i)$ which it can compute for itself) and forming the sum

$$S_i = \sum_{l=1}^{n} P_l(i) \bmod q$$

## Problem 2

How does a set of $k$ participants use their respective $S_i$ values to decode an ElGamal encryption $(X, Y)$?

# Problem 2

Lagrange Interpolation:

Given $k$ distinct pairs $(i, S_i)$ with $i \in I$, form the interpolated polynomial by computing

$$P(x) = \sum_{i \in I} \left( S_i \frac{\prod_{l \neq i \in I}(x - l)}{\prod_{l \neq i \in I}(i - l)} \right) \bmod q$$

The joint secret can be computed as

$$S = P(0) = \sum_{i \in I} \left( S_i \frac{\prod_{l \neq i \in I}(0 - l)}{\prod_{l \neq i \in I}(i - l)} \right) \bmod q$$

# Problem 2

$$S = P(0) = \sum_{i \in I} \left( S_i \frac{\prod_{l \neq i \in I} (0 - l)}{\prod_{l \neq i \in I} (i - l)} \right) \bmod q$$

Each $T_i$ with $i \in I$ can compute its own portion of the sum

$$Z_i = S_i \frac{\prod_{l \neq i \in I} (0 - l)}{\prod_{l \neq i \in I} (i - l)} \bmod q$$

$$S = P(0) = \sum_{i \in I} Z_i \bmod q$$

# Group ElGamal Encryption

- Each recipient selects a large random private key $a_i$ and computes an associated public key $A_i = g^{a_i} \bmod p$.
- The group key is $A = \prod A_i \bmod p = g^{\sum a_i} \bmod p$.
- To send a message $M$ to the group, Bob selects a random value $r$ and computes the pair $(X, Y) = (A^r M \bmod p, g^r \bmod p)$.
- To decrypt, each group member computes $Y_i = Y^{a_i} \bmod p$. The message $M = X / \prod Y_i \bmod p$.

## Problem 2

$$S = P(0) = \sum_{i \in I} Z_i \mod q$$

Each $T_i$ with $i \in I$ computes $V_i = Y^{Z_i} \bmod p$.

The ElGamal encryption $(X, Y)$ can now be decrypted as $X / \prod_{i \in I} V_i \bmod p$.

## Problem 3

Given a set of ElGamal encryptions $(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)$, create an equivalent set of ElGamal encryptions and prove the equivalence without revealing the correspondence.

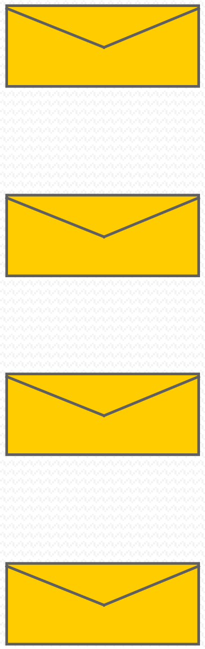
# Problem 3

Use ElGamal re-encryption to create new encryptions of the same values $(X_1', Y_1'), (X_2', Y_2'), \ldots, (X_n', Y_n')$ and permute the results to create a new set.

Interactively prove the equivalence by creating, say, 100 additional equivalent permuted "intermediate" sets in exactly the same way.

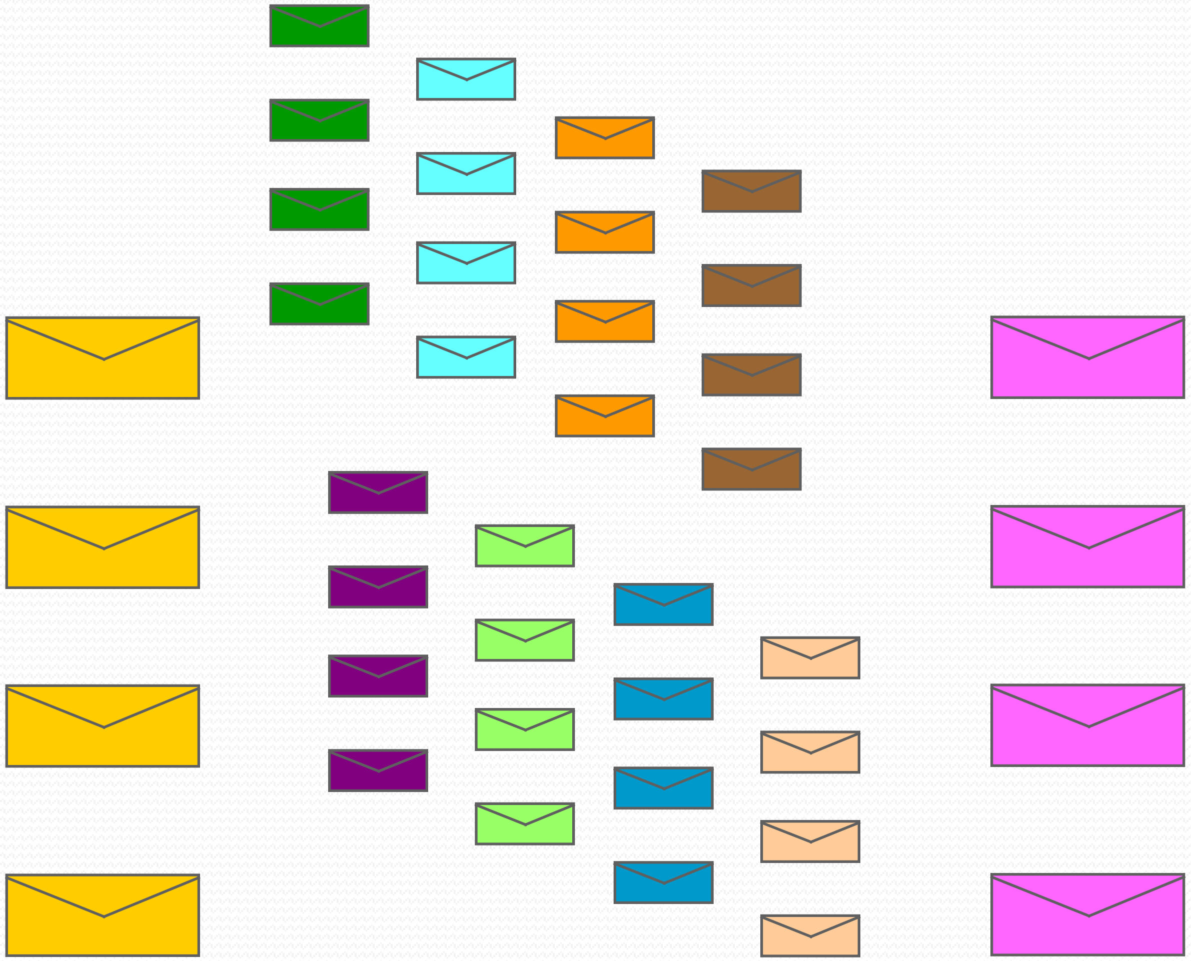
Answer each challenge by associating each intermediate set with either the original set of the new derived set.

## A Verifiable Re-encryption Mix

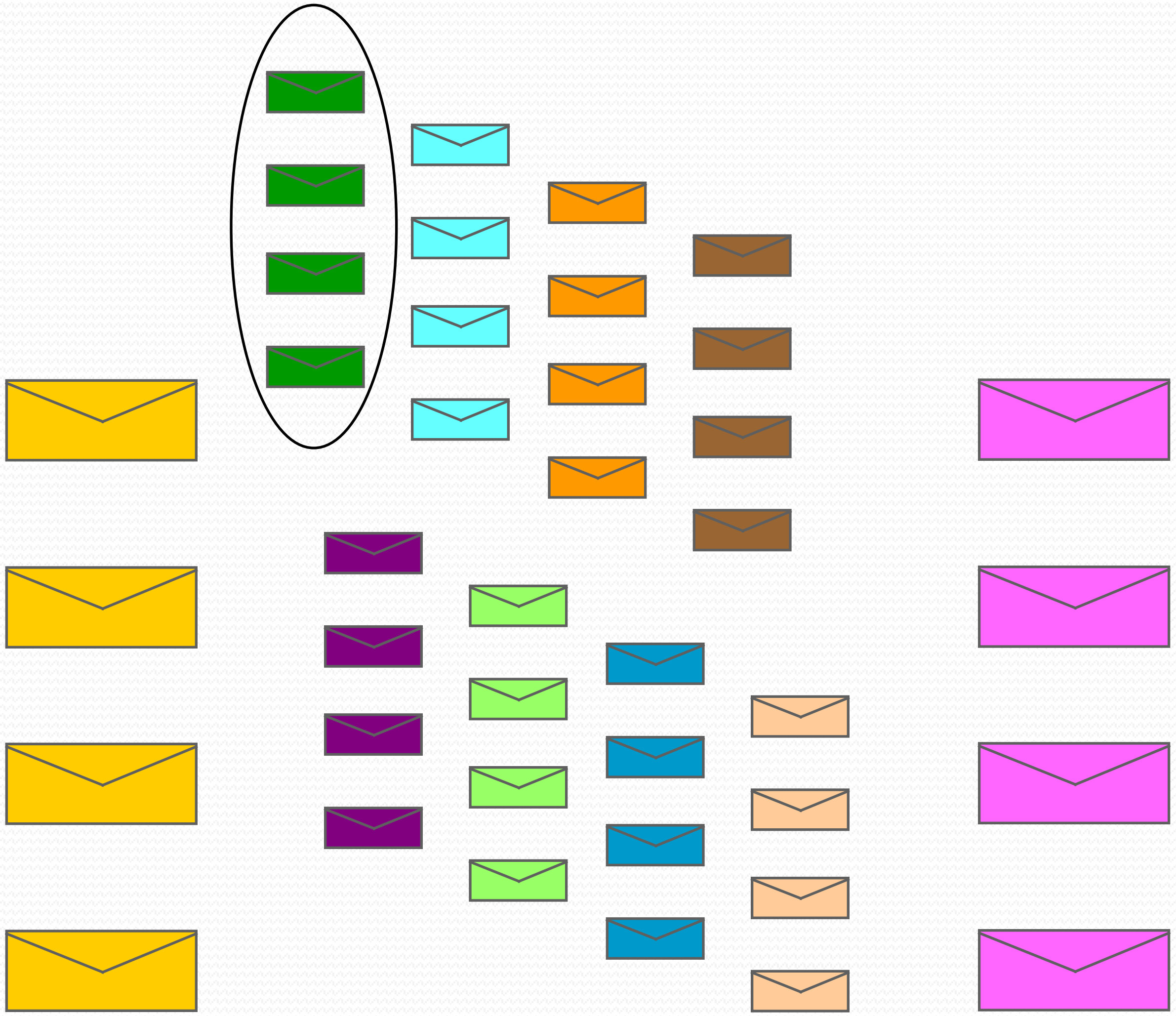


## A Verifiable Re-encryption Mix

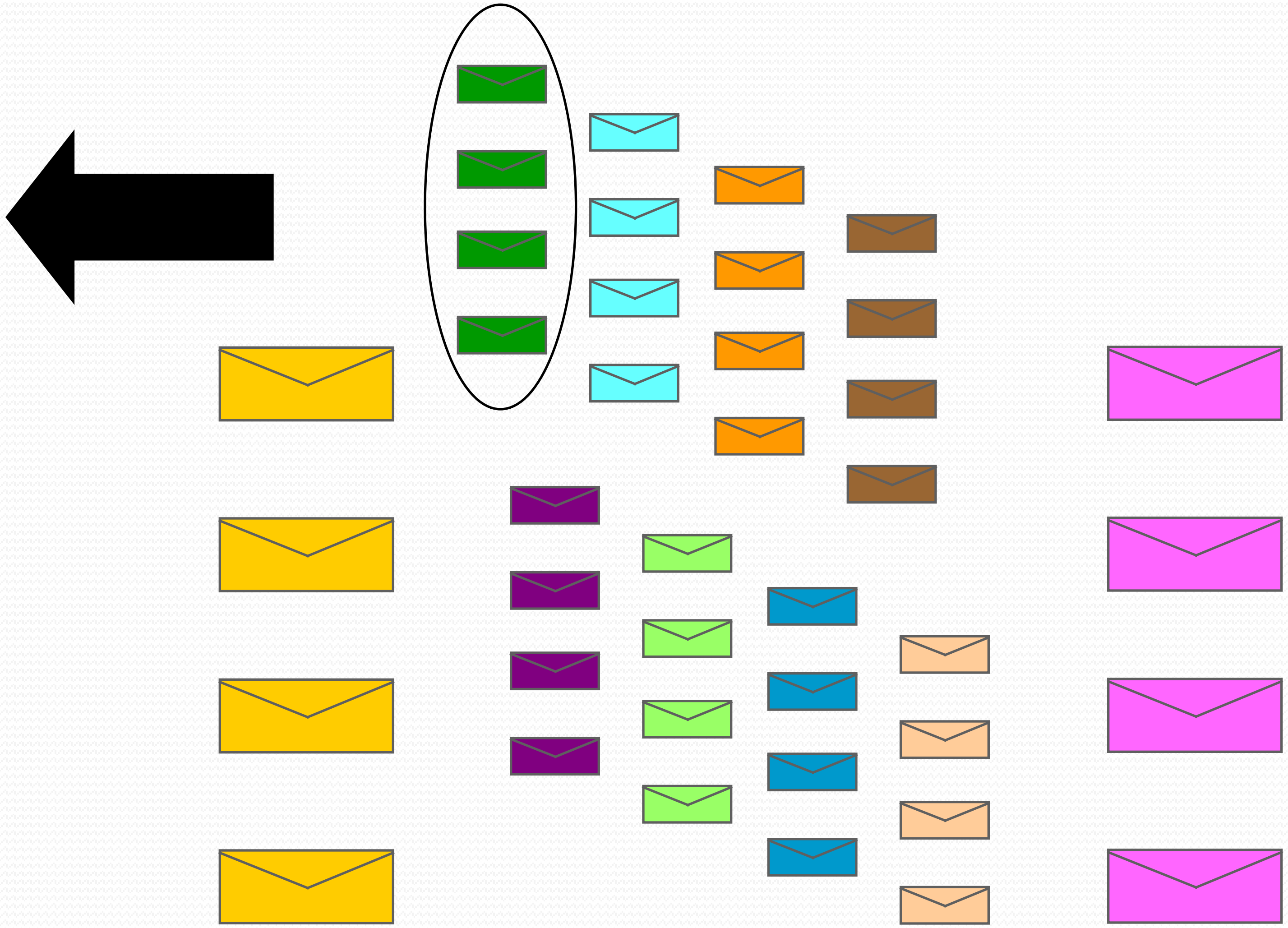
## A Verifiable Re-encryption Mix



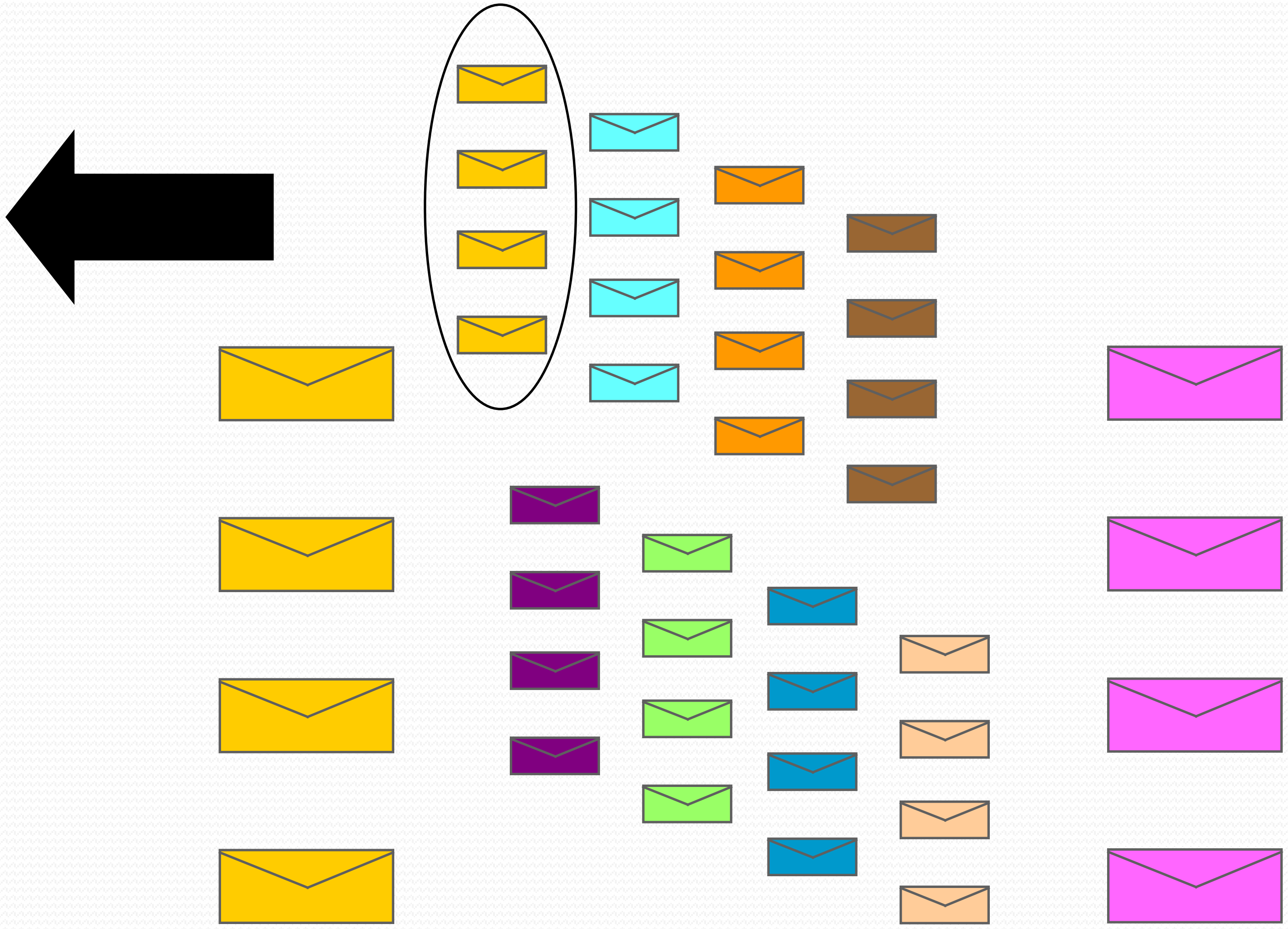
# A Verifiable Re-encryption Mix



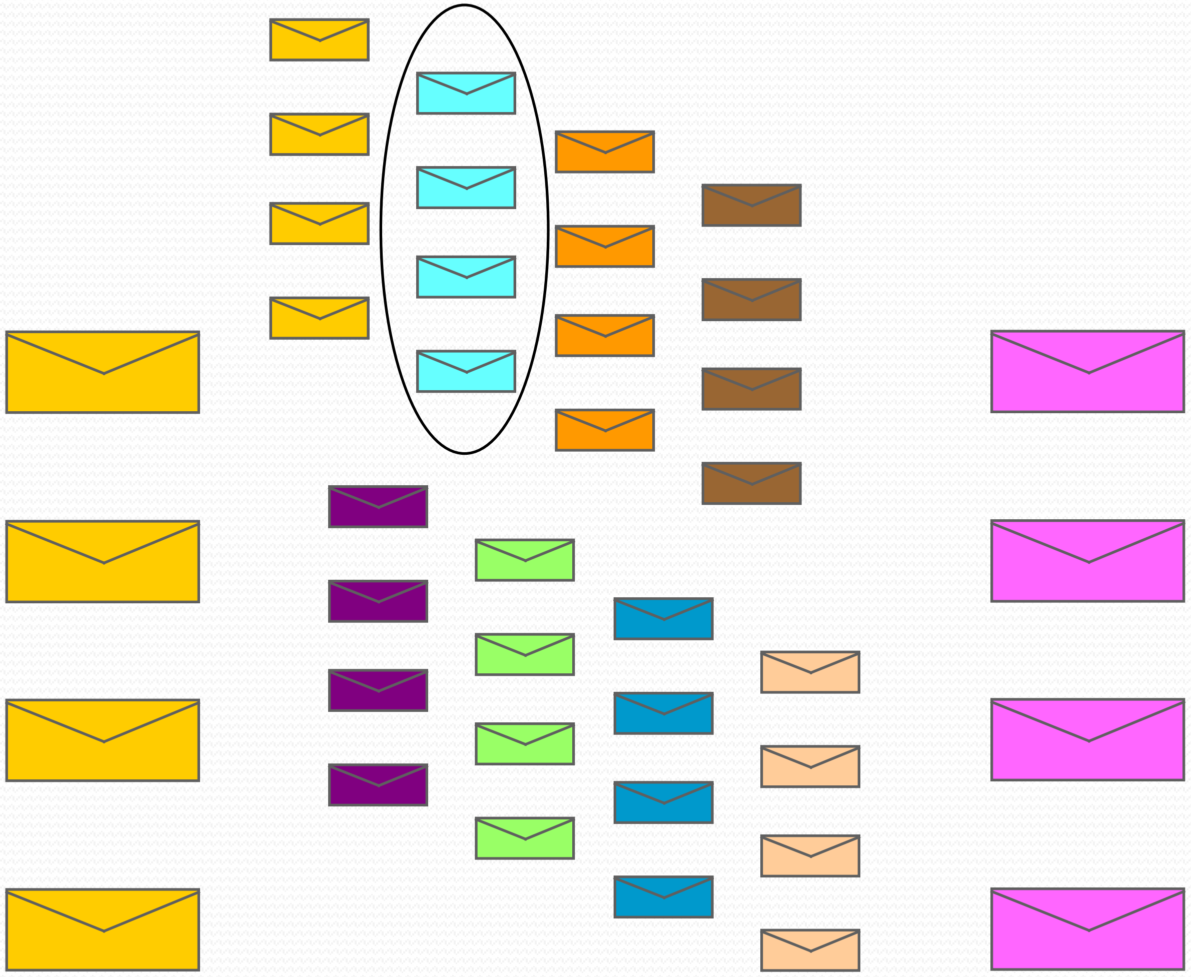
## A Verifiable Re-encryption Mix



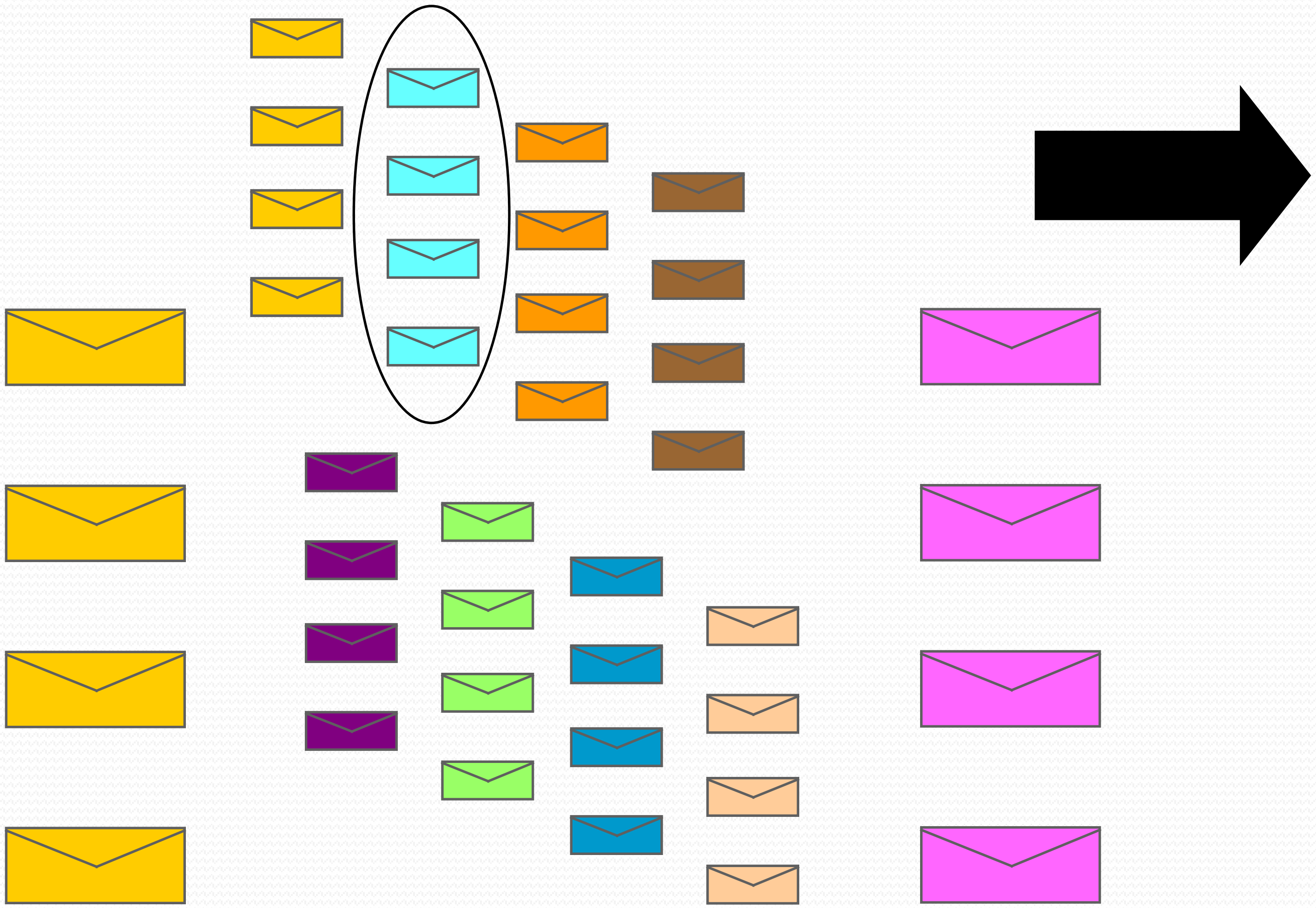
# A Verifiable Re-encryption Mix



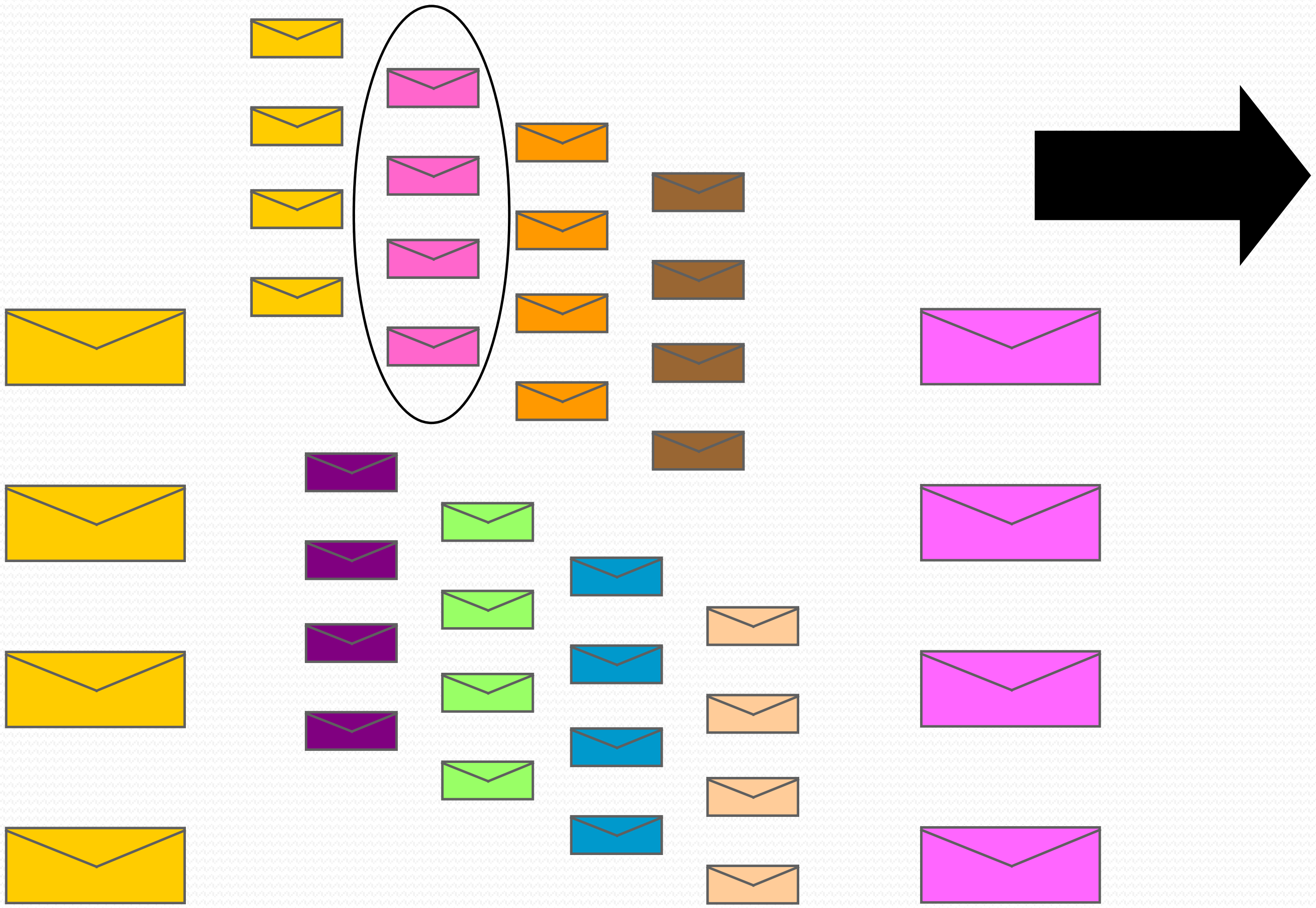
# A Verifiable Re-encryption Mix



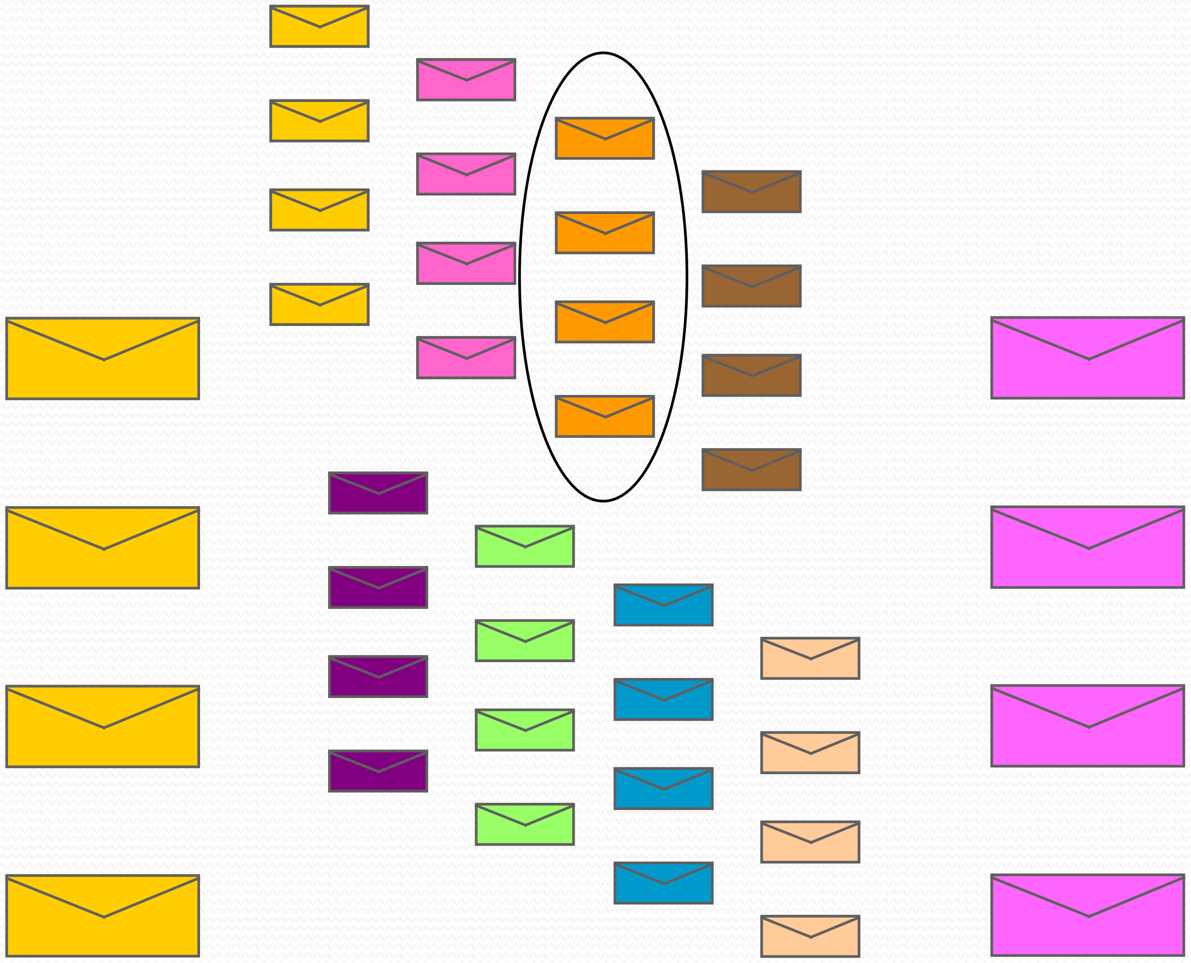
## A Verifiable Re-encryption Mix



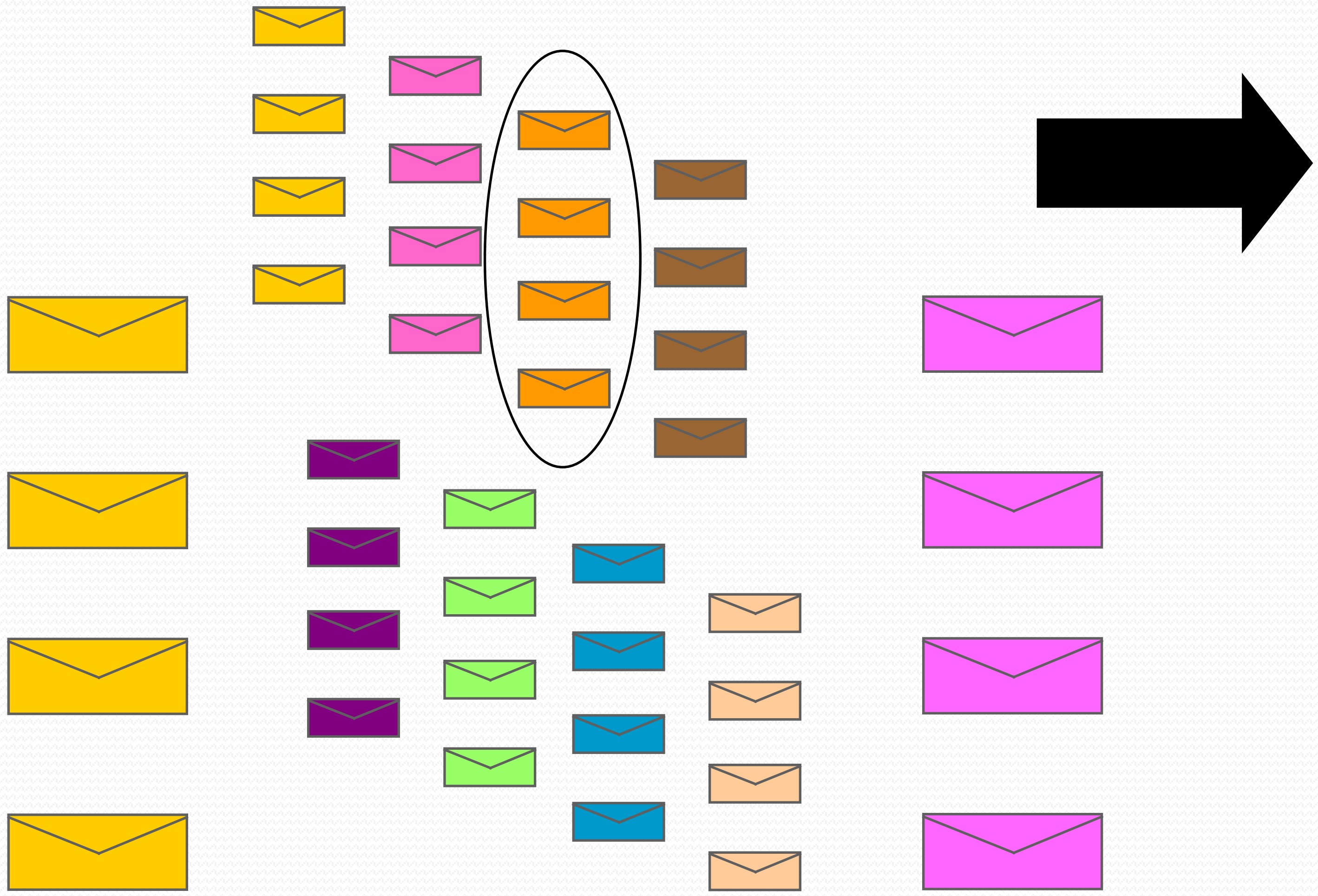
# A Verifiable Re-encryption Mix



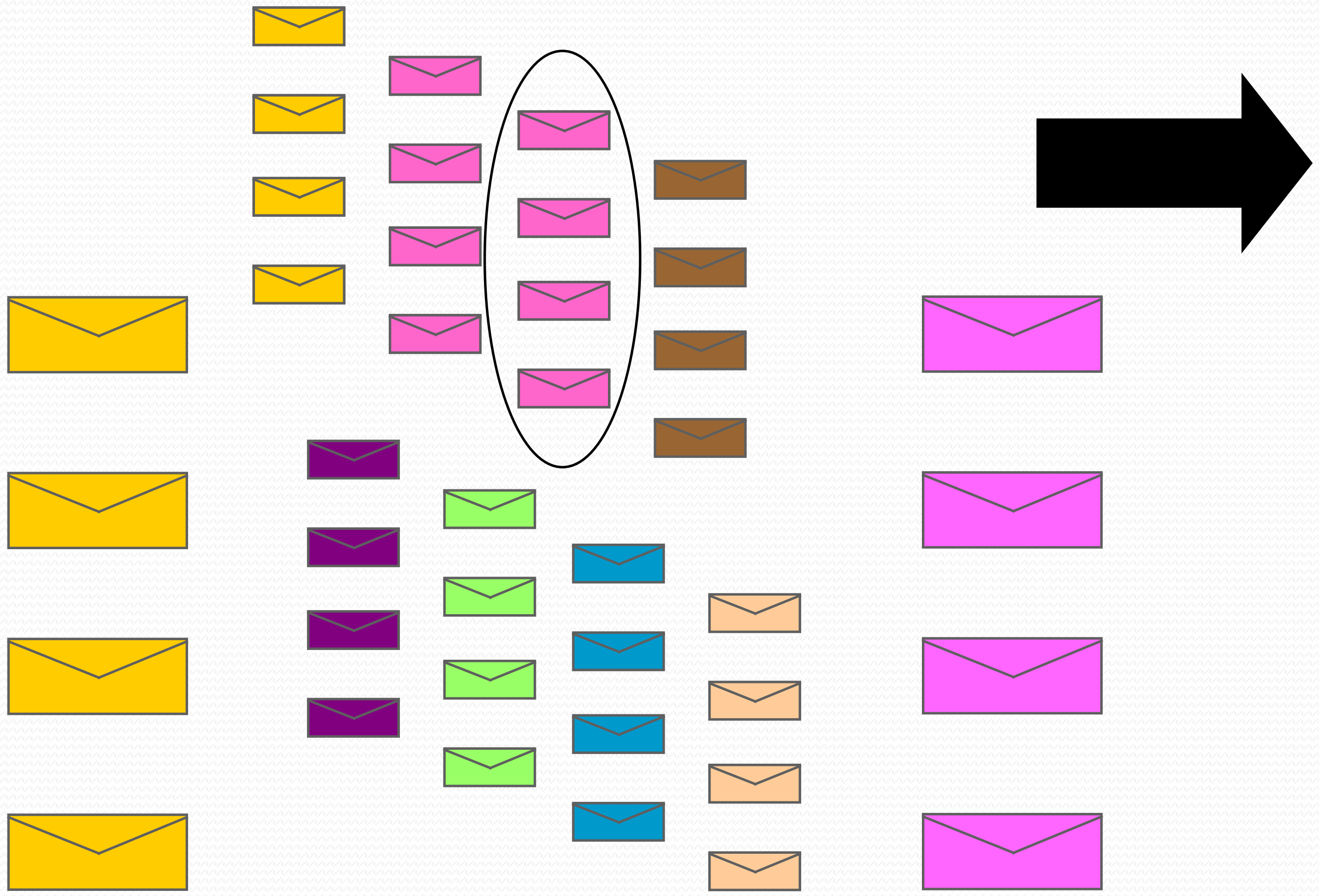
## A Verifiable Re-encryption Mix



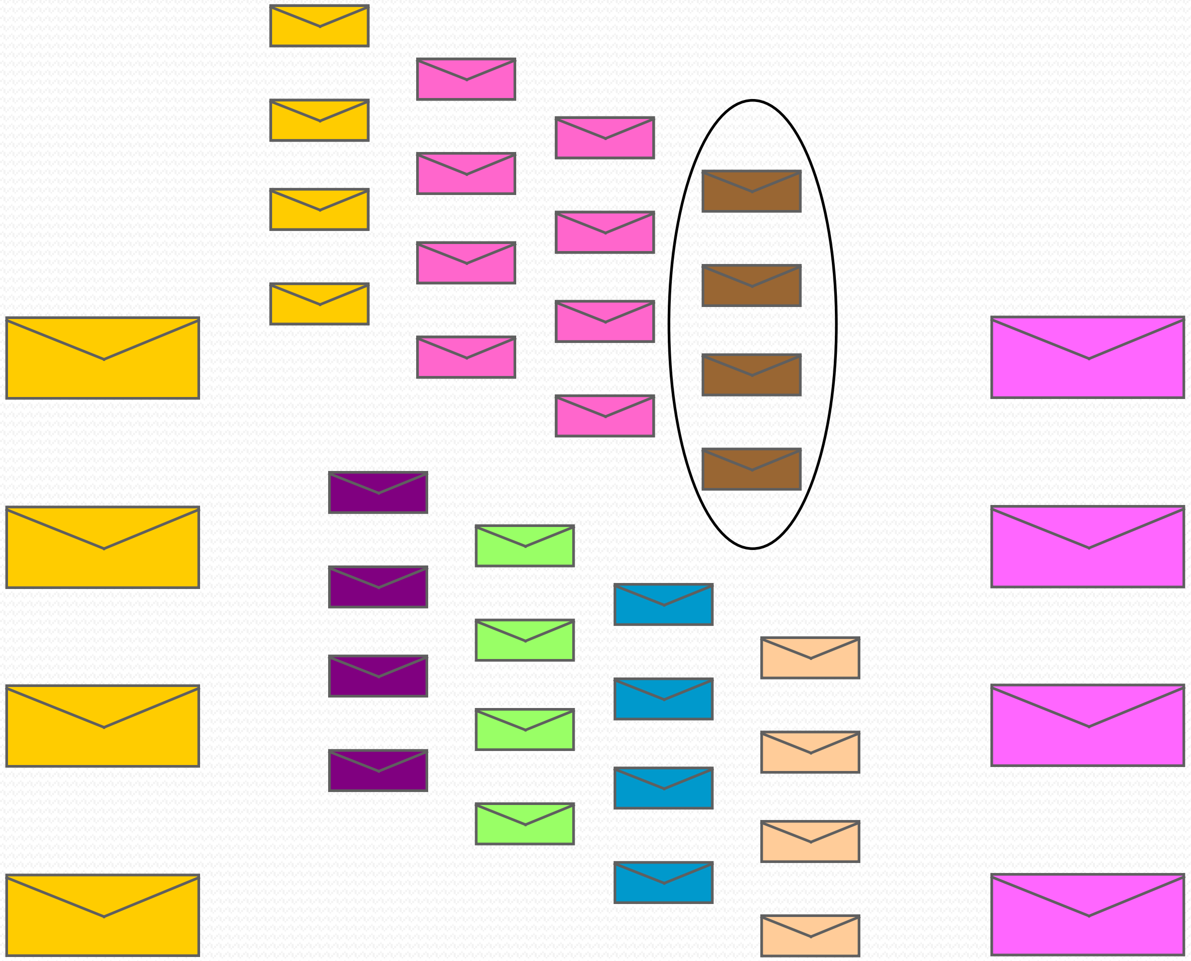
## A Verifiable Re-encryption Mix



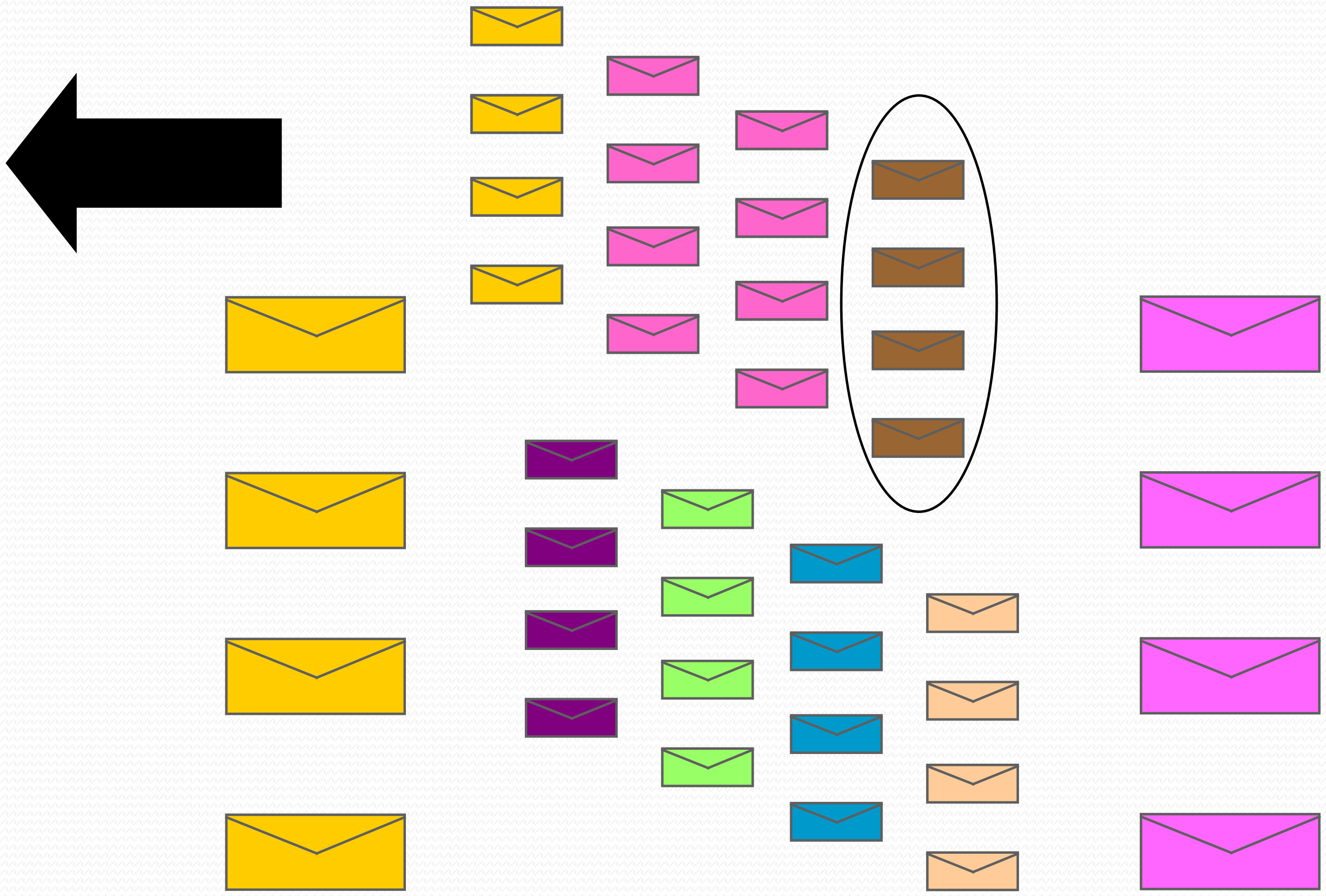
# A Verifiable Re-encryption Mix



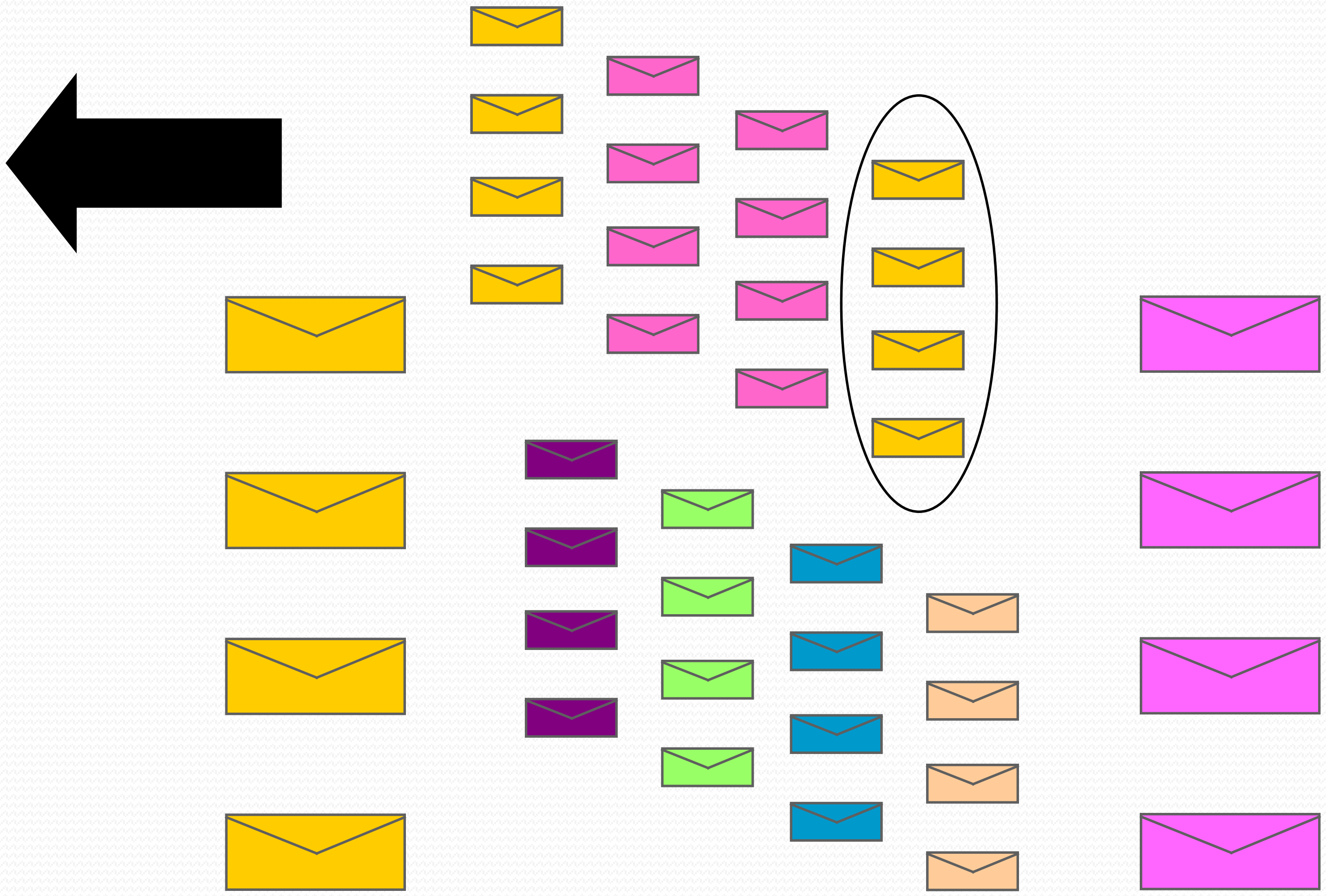
## A Verifiable Re-encryption Mix



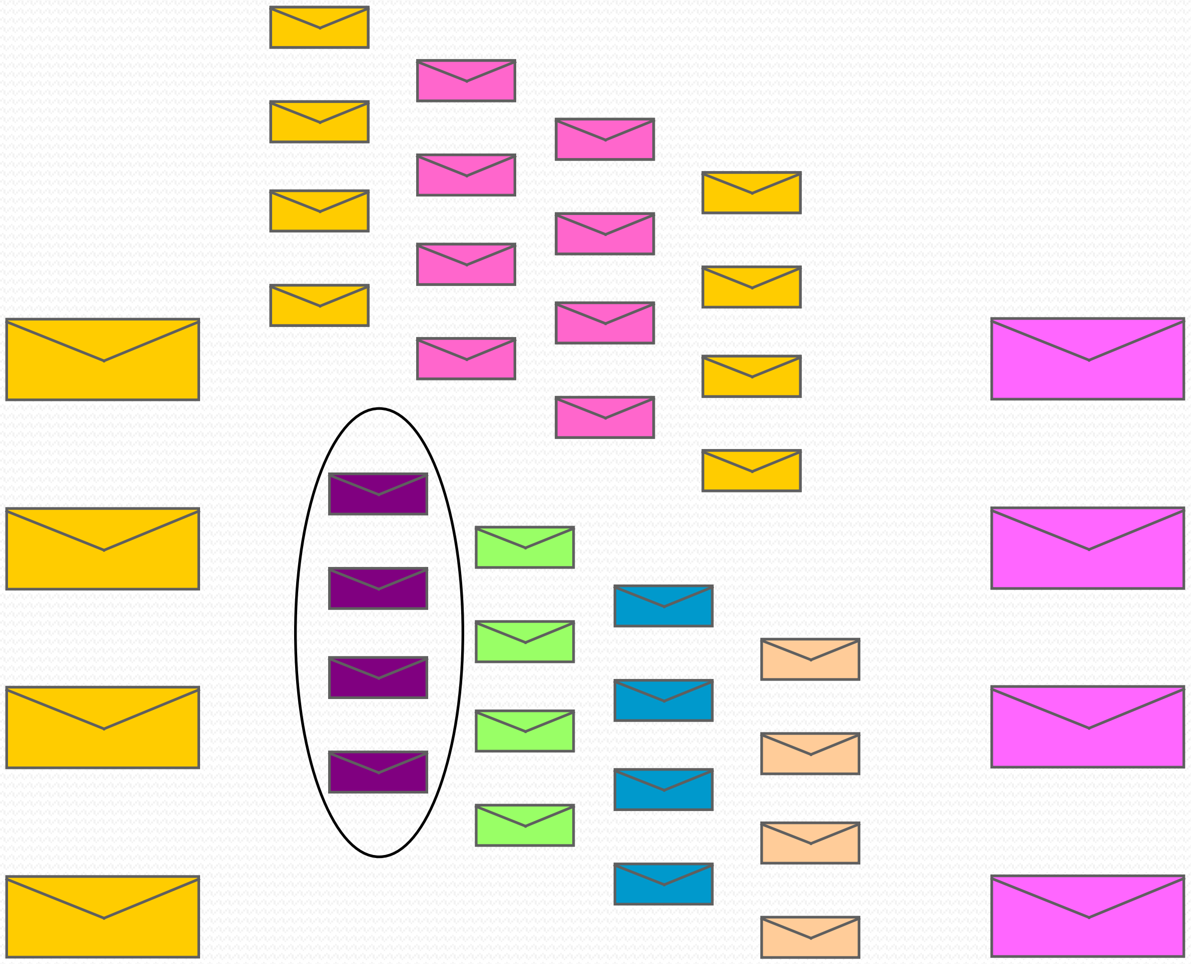
## A Verifiable Re-encryption Mix



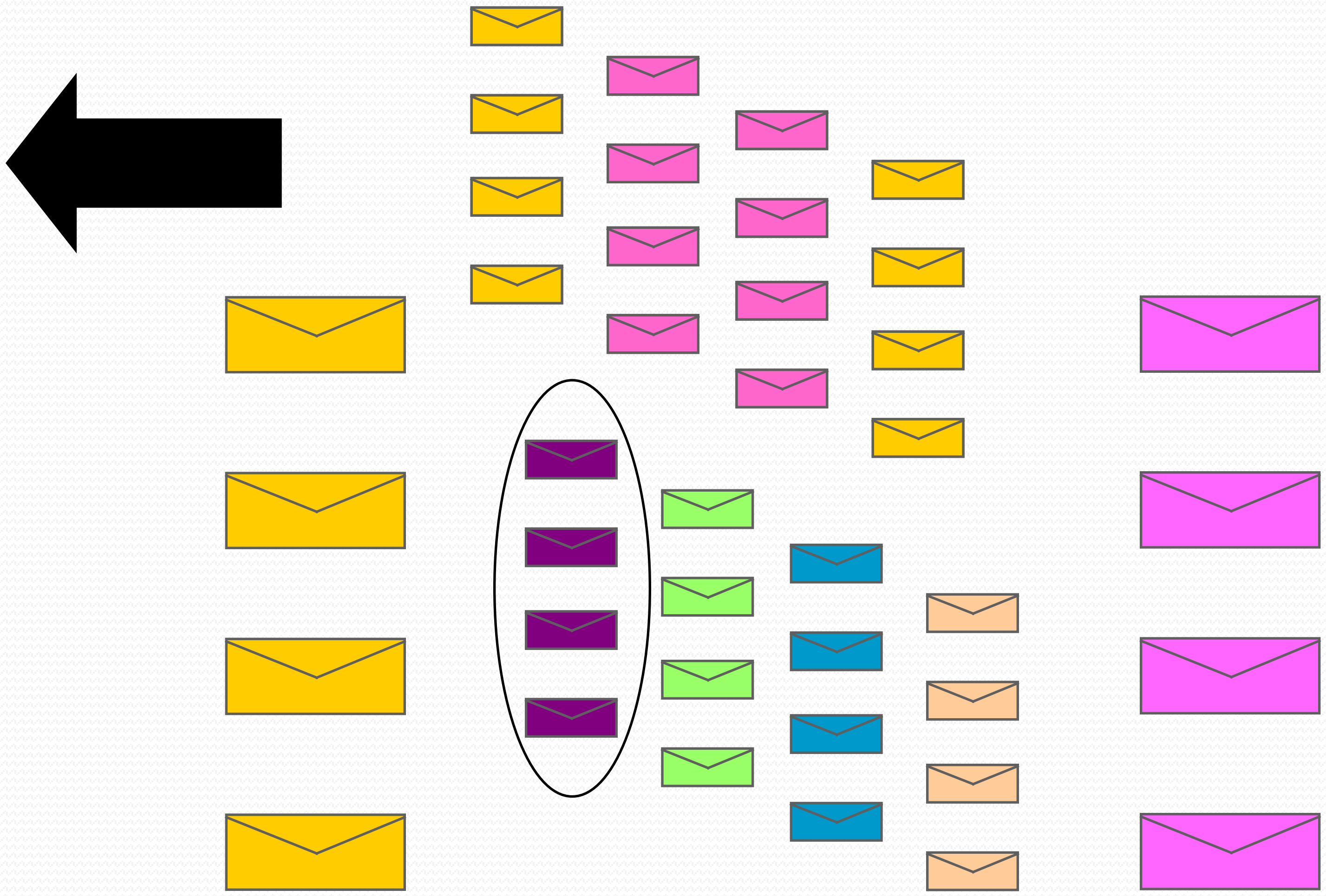
## A Verifiable Re-encryption Mix



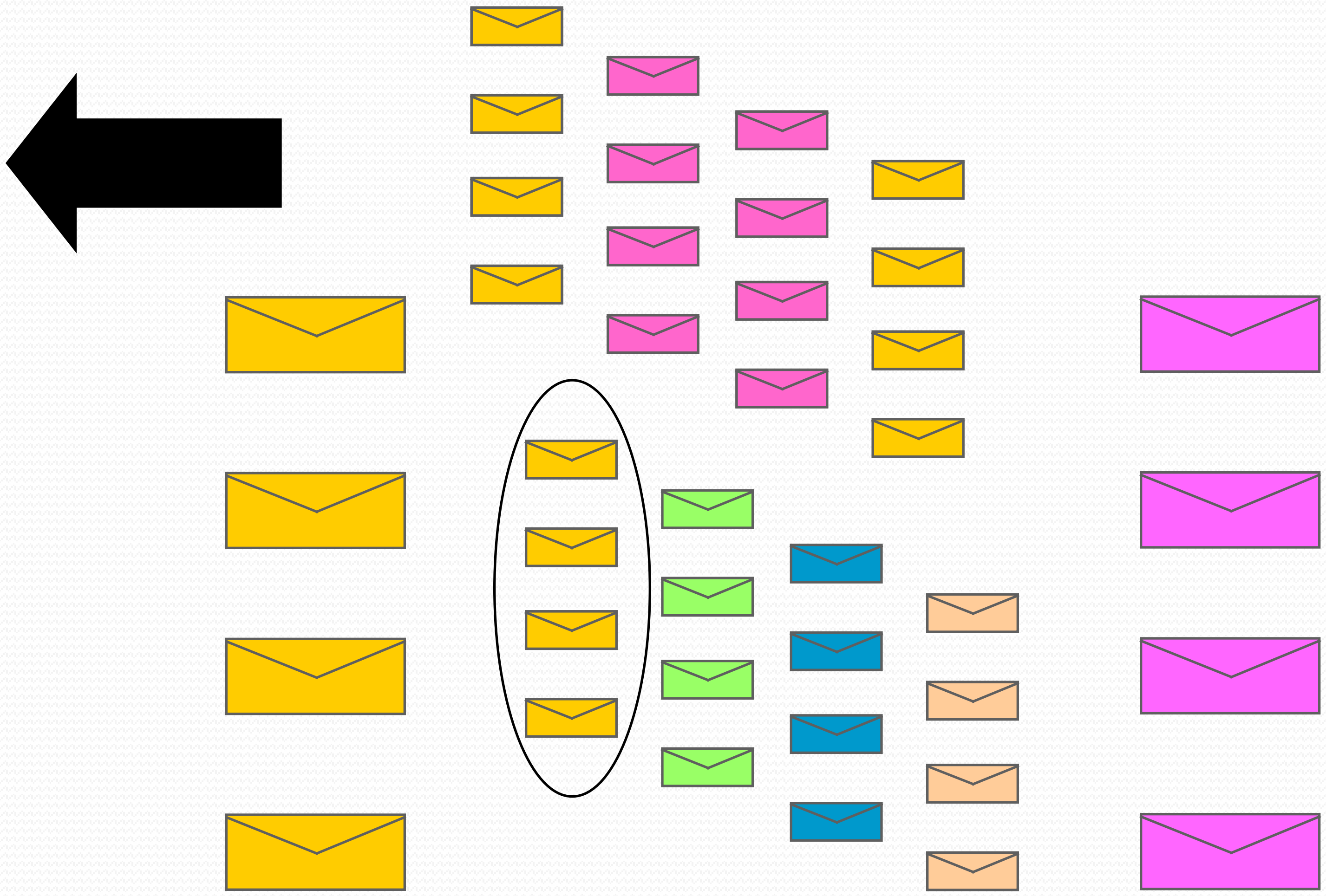
## A Verifiable Re-encryption Mix



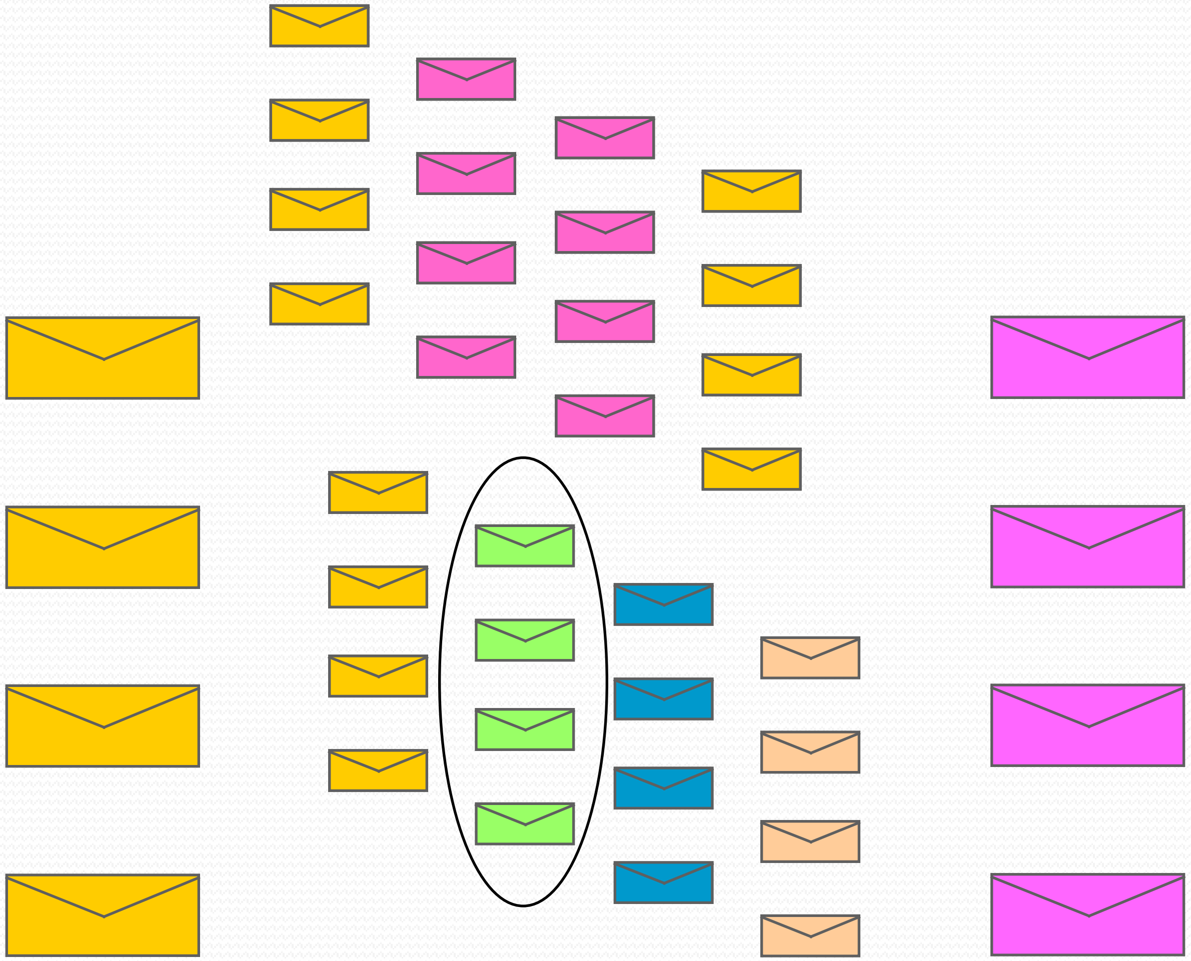
## A Verifiable Re-encryption Mix



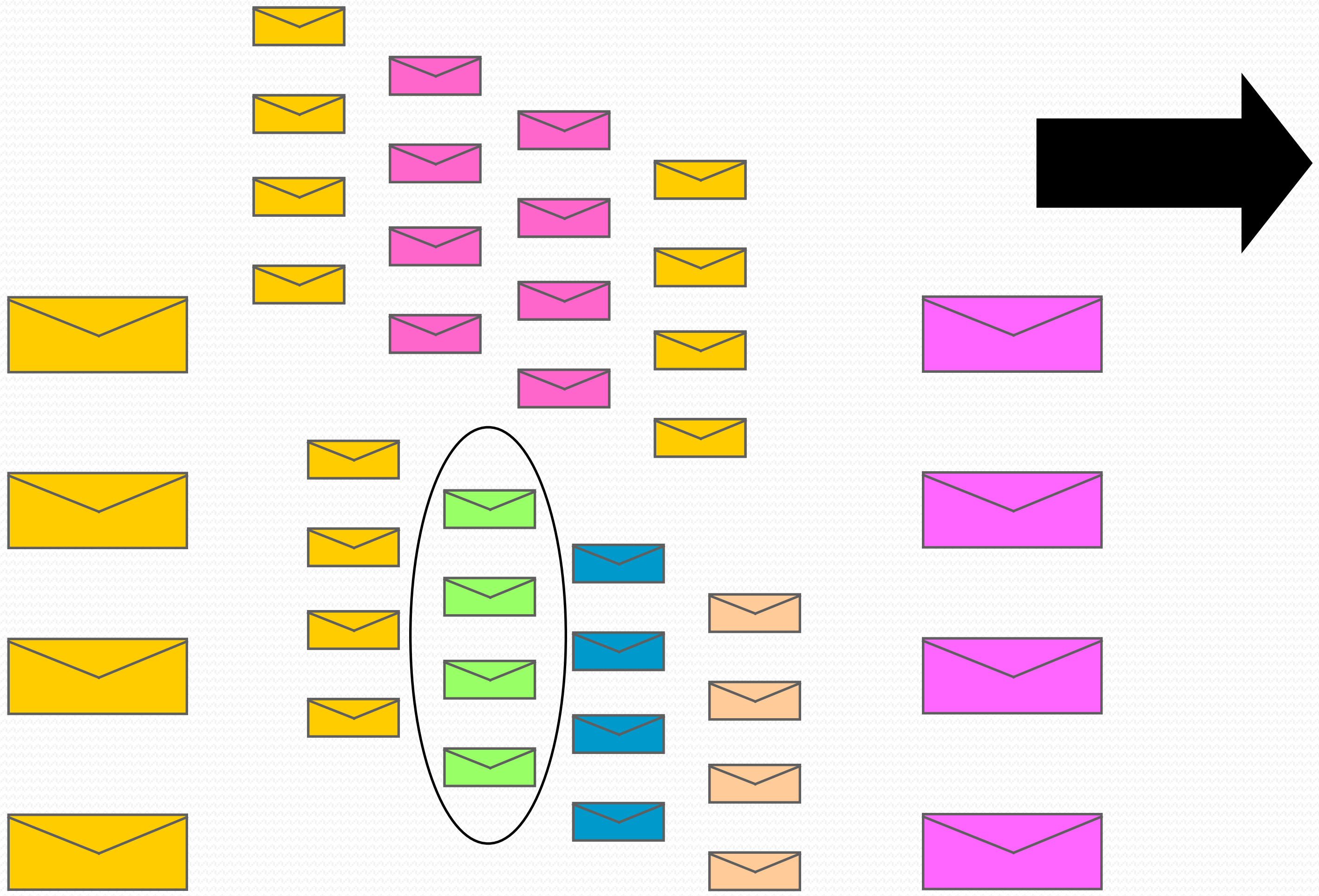
# A Verifiable Re-encryption Mix



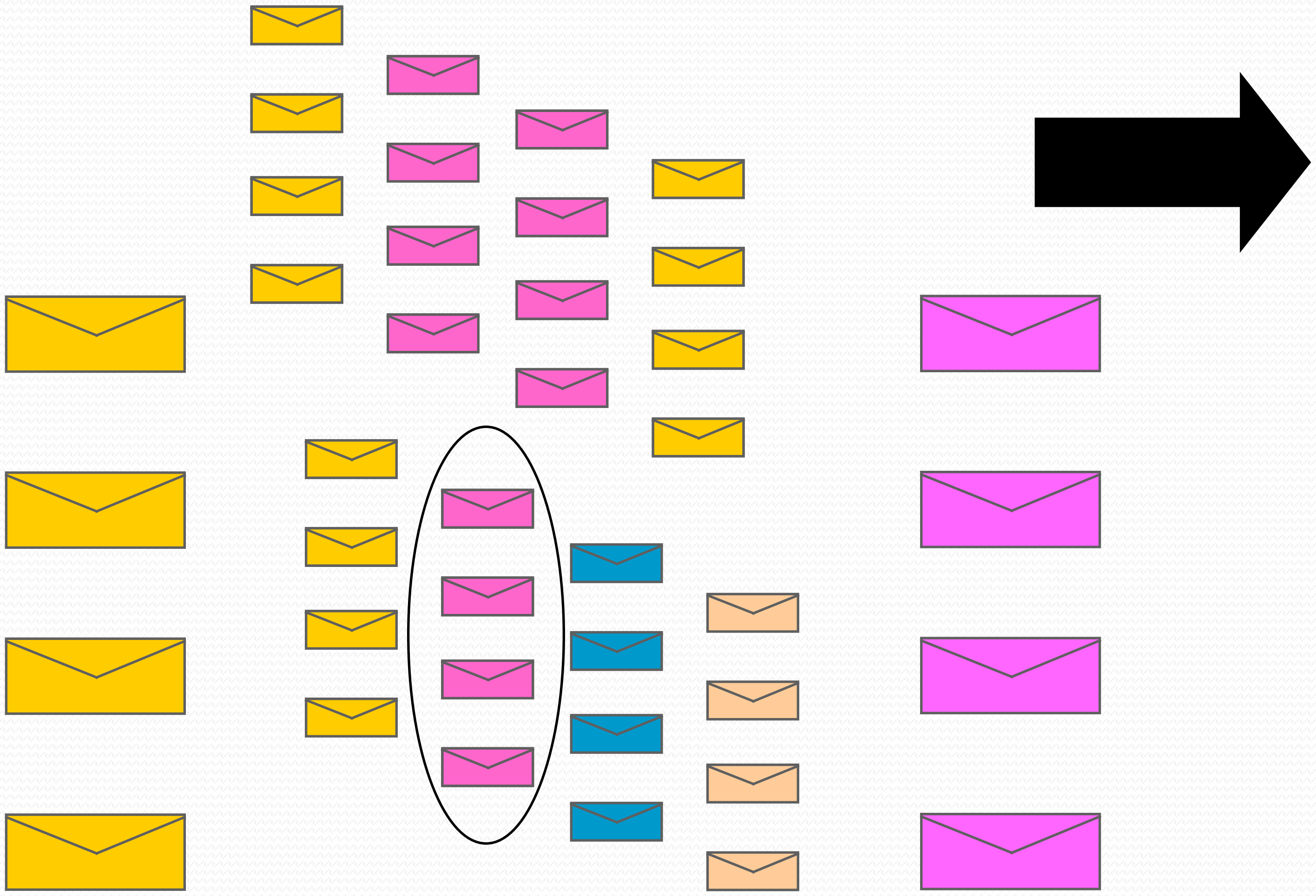
## A Verifiable Re-encryption Mix



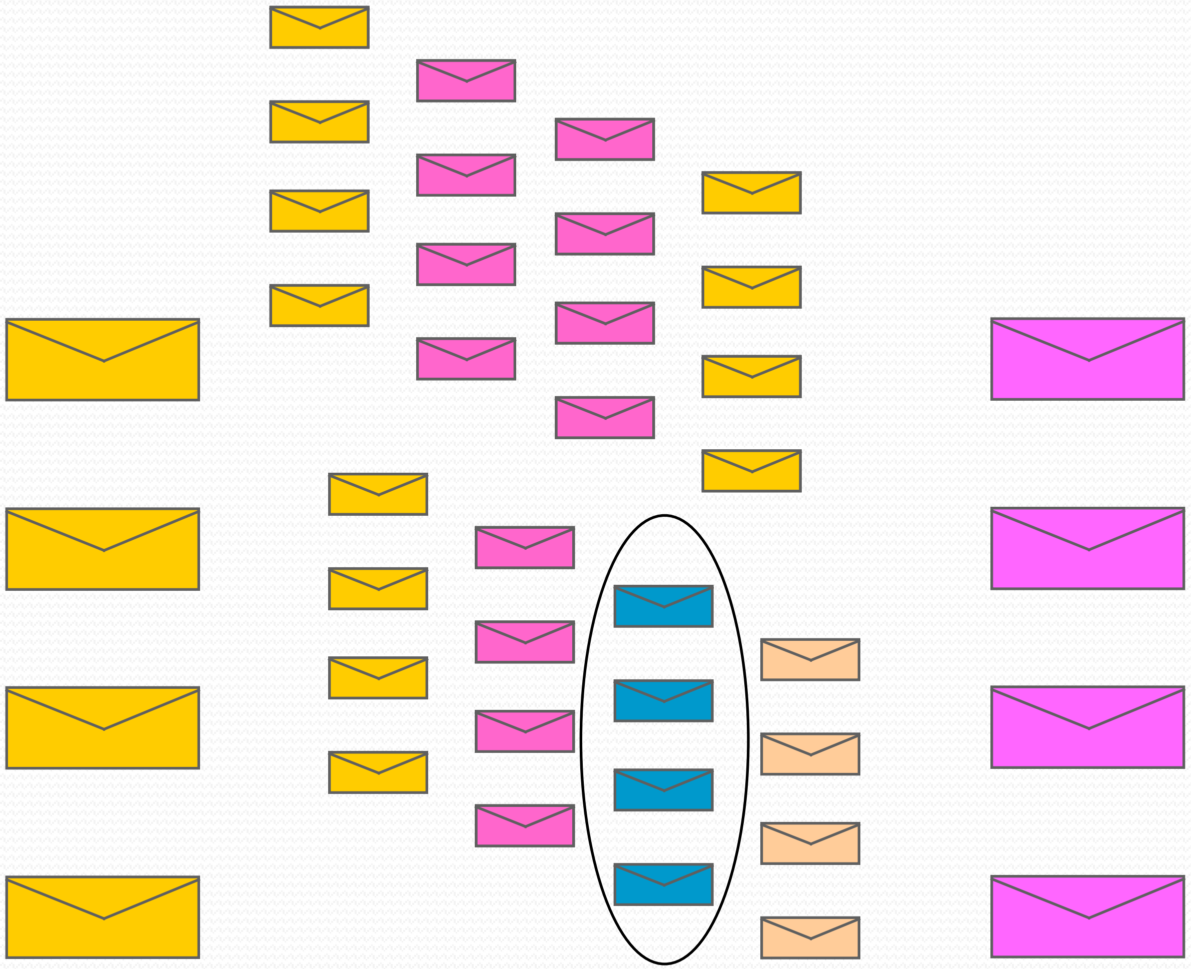
# A Verifiable Re-encryption Mix



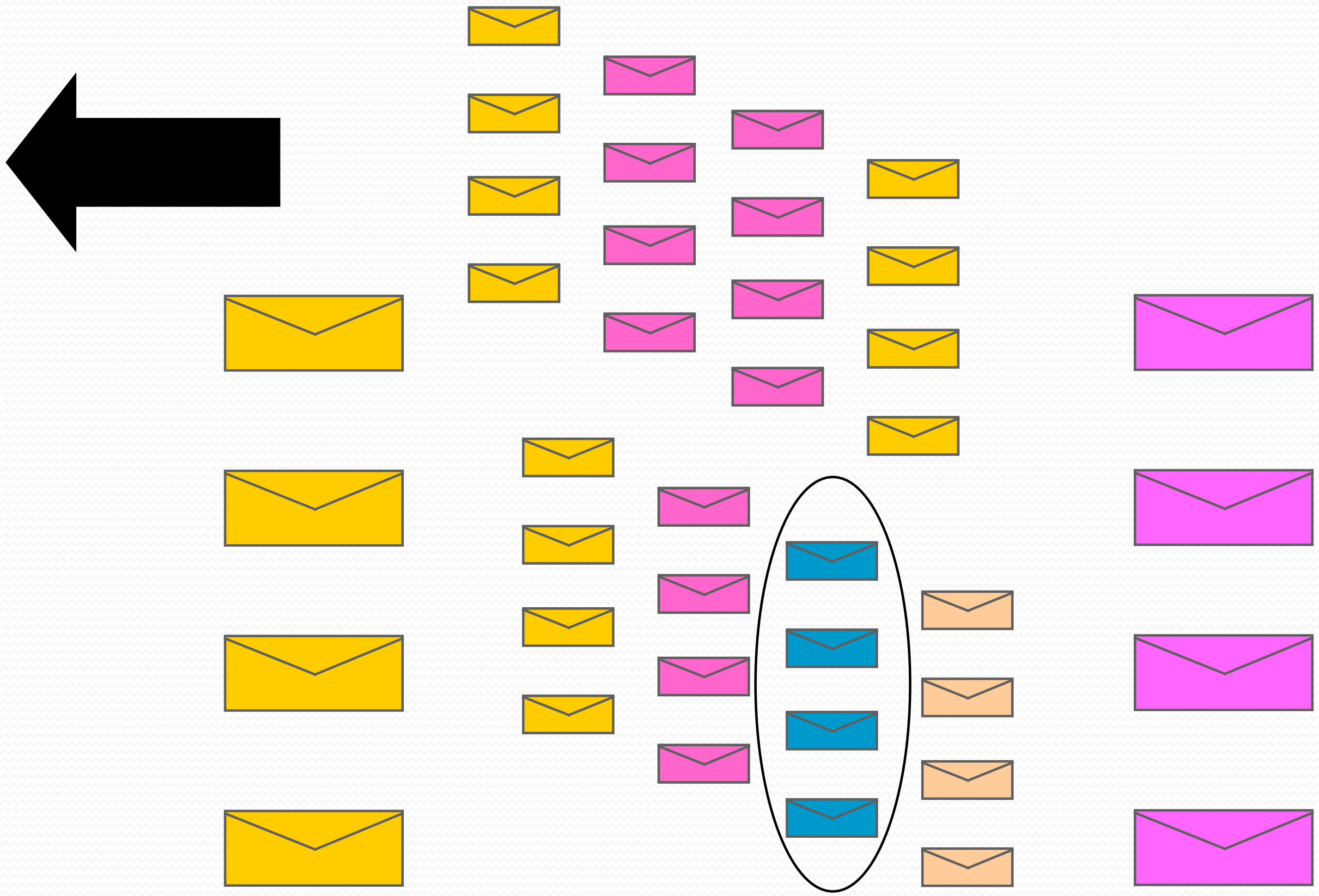
## A Verifiable Re-encryption Mix



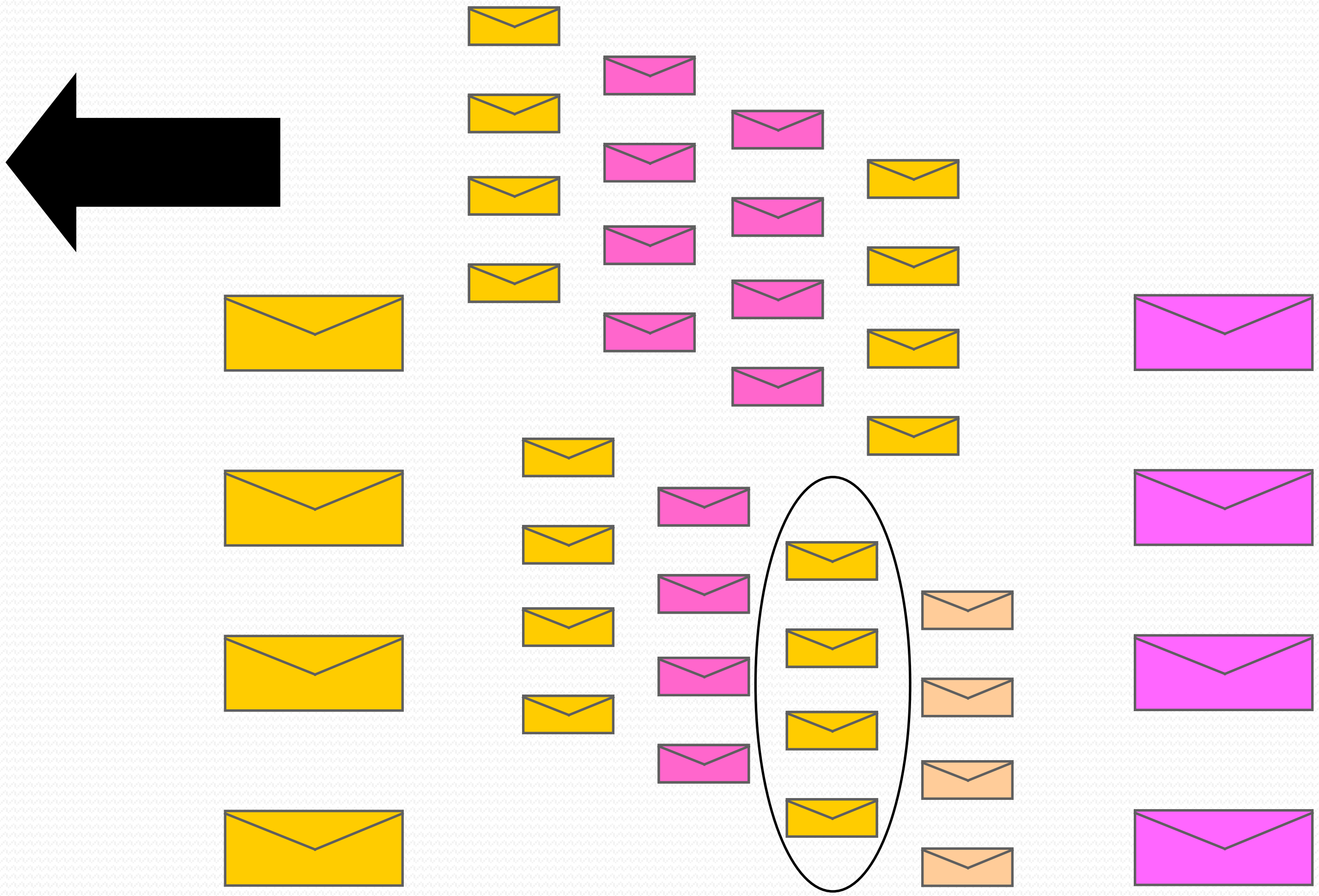
# A Verifiable Re-encryption Mix



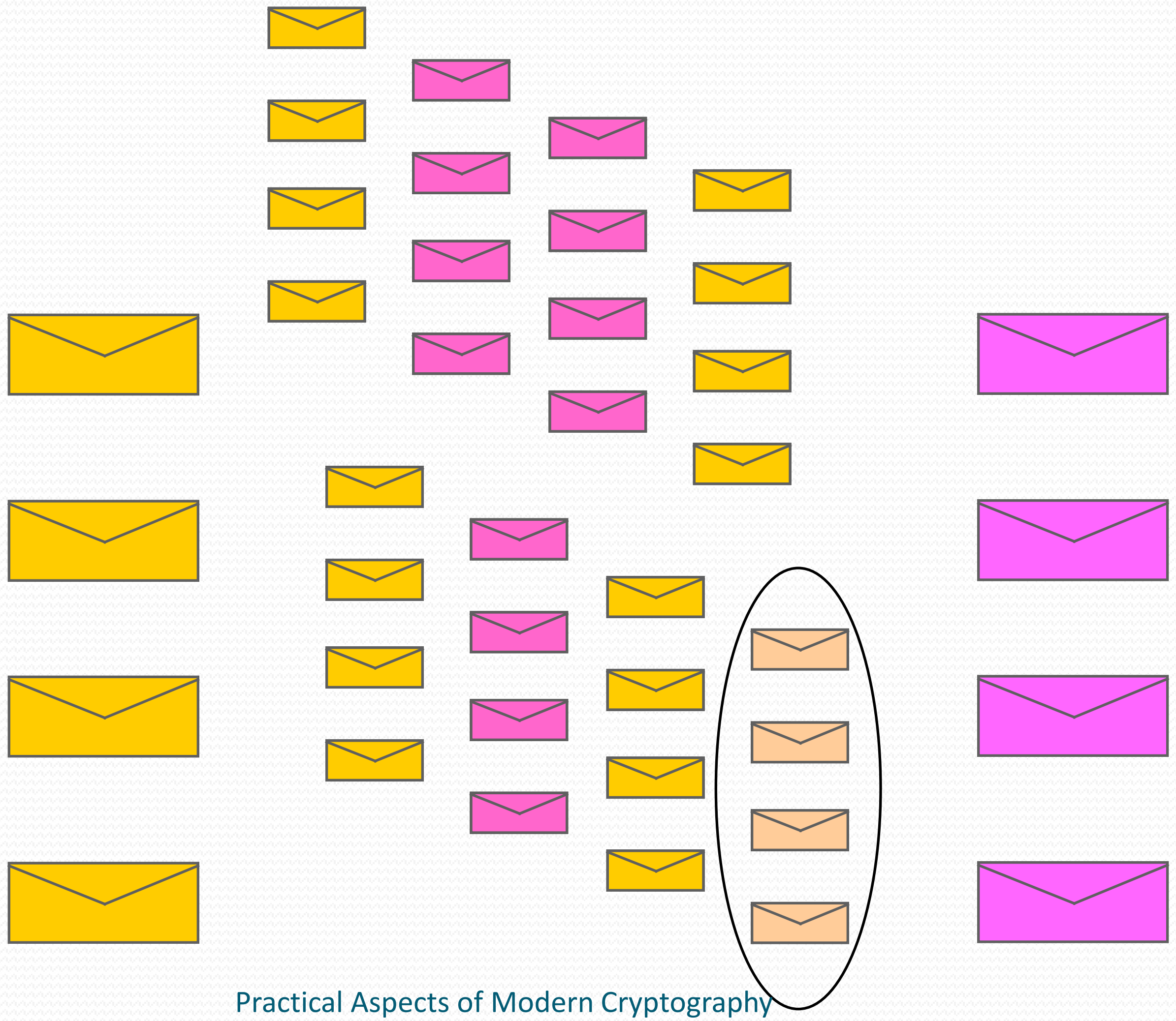
## A Verifiable Re-encryption Mix



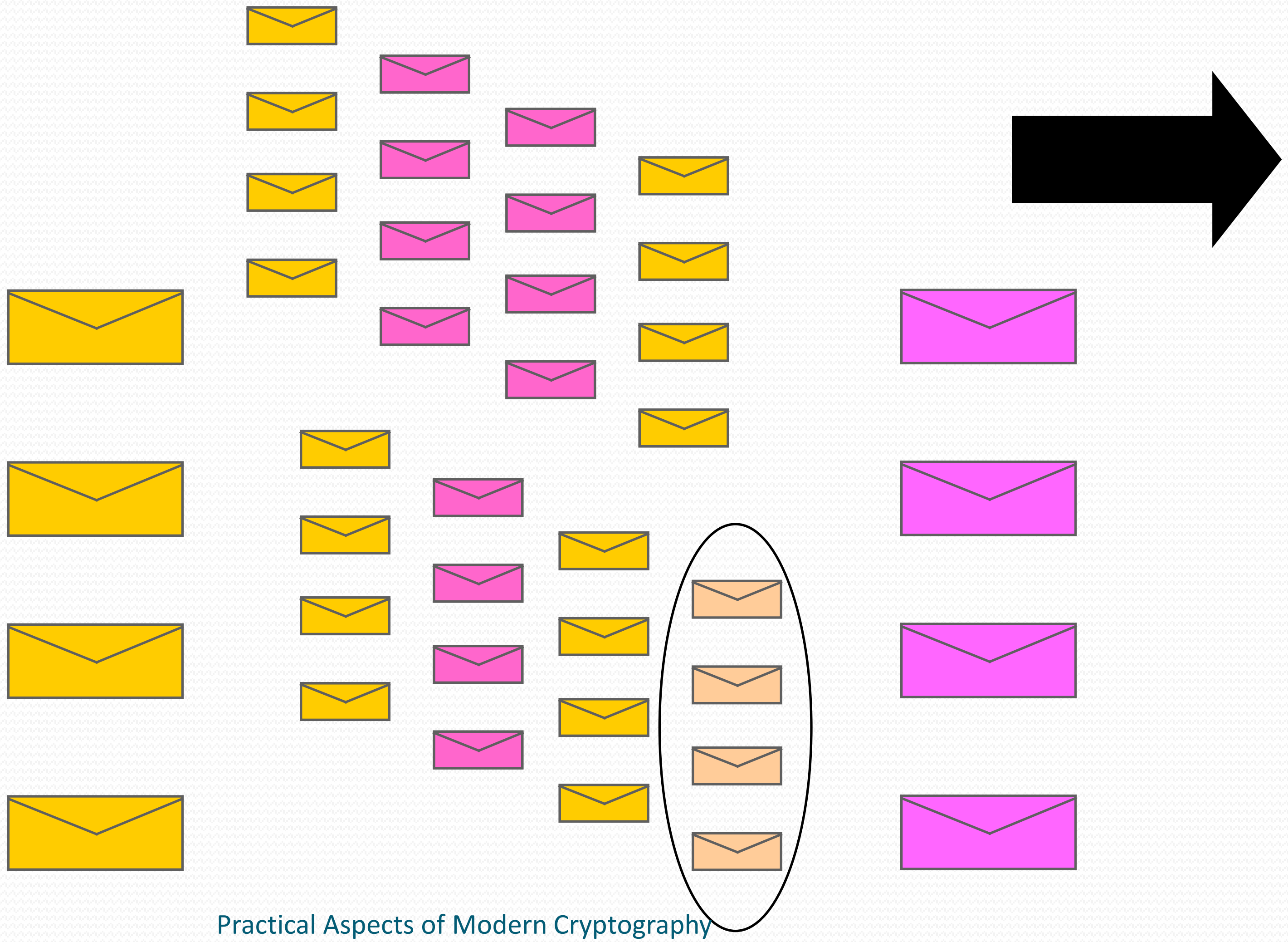
# A Verifiable Re-encryption Mix



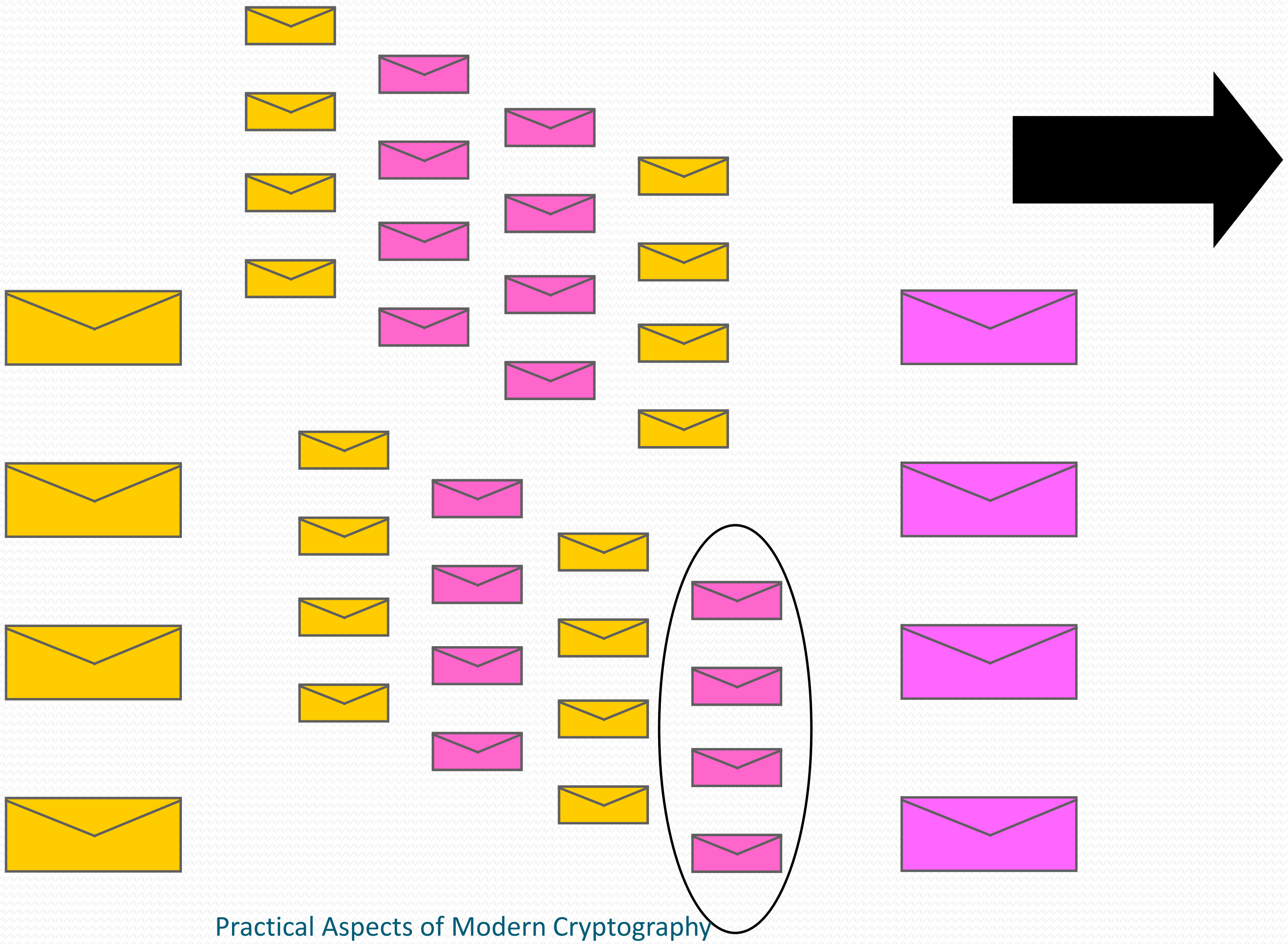
# A Verifiable Re-encryption Mix



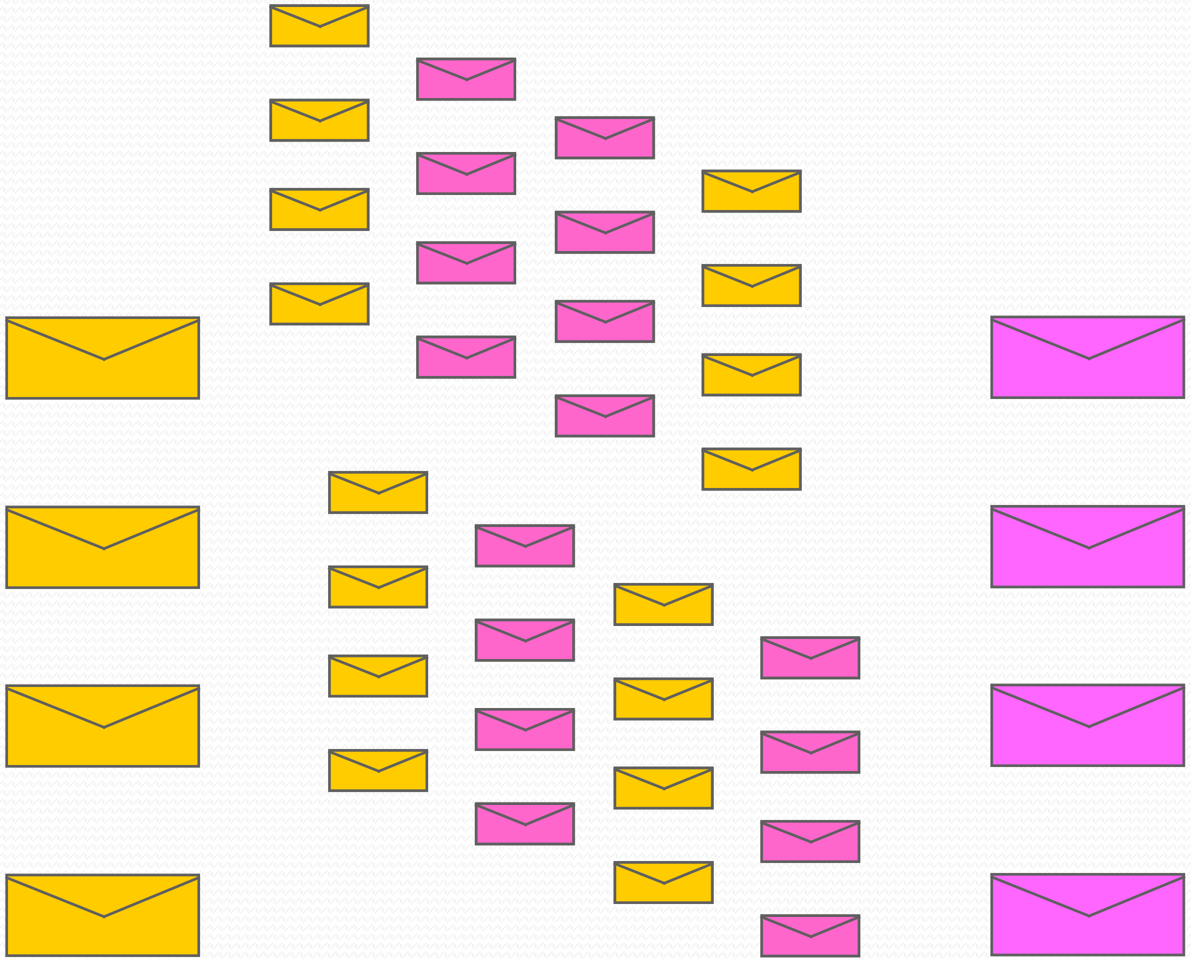
# A Verifiable Re-encryption Mix



# A Verifiable Re-encryption Mix



# A Verifiable Re-encryption Mix



## Problem 3

The challenges for this re-encryption mix can be obtained by feeding all of the intermediate and final ballot sets into a cryptographic hash function such as SHA-1.

The output bits of the hash can be used as the challenge bits in an interactive proof.