# Supplementary Material

There is a need for a method for sharing collections of products on the internet by means of an image, where the original link associated with the image in a web-browser can be restored if it has been modified or removed due to subsequent re-sharing. In this section, we present a method to provide an improved system for sharing collections of products over the internet, where the original link associated with the image in a web-browser can be restored if it has been modified or removed. This digital process receives a collection of online products, and associates the collection with a unique identifier. The unique identifier is encoded and engrained into the pixels and the metadata of a "hook" image representing the collection of products. The reference to the collection can later be recovered by decoding the unique identifier engrained in the "hook" image, and the link on the image updated to maintain the correct reference to the collection even after the image has been shared on social networks. In this section, we present a technology to embed images with a unique identifier (OPCI) that allows for the original product collection link to be reconstructed if it were lost. We will describe next how we encode the binary number into the stamp and make the stamp part of the image.

**Stamp composition**

Let the "hook" image height be $I_h$ and the image width be $I_w$. Without loss of generality, assume that $I_h > I_w$. Let $S_w$ be the stamp width, $S_h$ be the stamp height, $S_p$ the ratio of the stamp width to the image width, $Cross_{prop}$ be the proportion of the plus width to the stamp width, $C$ be a component and a square of pixels in the stamp outside the plus area, $Cross_h$ the height of the plus sign, $Cross_w$ the width of the plus sign, $Q^1$ the quadrant on the top left corner of the stamp bounded by the plus sign and the stamp border, $Q^2$ the quadrant on the top right of the stamp bounded by the plus sign and the stamp border, $Q^3$ the quadrant on the lower left of the stamp bounded by the plus sign and the stamp border, and $Q^4$ the quadrant on the lower right of the stamp bounded by the plus sign and the stamp border. Let $Q^n_w$ be the width of quadrant $n$, $Q^n_h$ be the height of quadrant $n$, $S_{padd_y}$ the size in pixels between the stamp and the bottom of an image, $S_{padd_x}$ the size in pixels between the stamp and the right edge of an image, $S_{center}$ the $(x, y)$ coordinate of the center of stamp with respect to the image's top left corner as origin, and $Q^n_{center}$ the center point of quadrant $n$ with respect to the image. More formally, the above

variables are calculate by,

$$S_p = (0, 1] \tag{1}$$
$$S_w = I_w * S_p \tag{2}$$
$$S_h = S_w \tag{3}$$
$$Cross_{prop} = (0, 1] \tag{4}$$
$$C^n_{ij-w} = \frac{Q^n_w}{4} \tag{5}$$
$$C^n_{ij-h} = \frac{Q^n_h}{4} \tag{6}$$
$$Cross_h = S_h * Cross_{prop} \tag{7}$$
$$Cross_w = S_w * Cross_{prop} \tag{8}$$
$$Q^n_w = \frac{Cross_w}{2} - \frac{Cross_{thick-x}}{2} - Cross_{padd-x} \tag{9}$$
$$Q^n_h = \frac{Cross_h}{2} - \frac{Cross_{thick-y}}{2} - Cross_{padd-y} \tag{10}$$
$$\tag{11}$$

$$S_{center} = \left(\left[I_w - S_{padd-x} - \frac{S_w}{2}\right], \left[I_h - S_{padd-y} - \frac{S_h}{2}\right]\right) \tag{12}$$

$$Cross_{center} = S_{center} \tag{13}$$

$$Q^1_{center} = (x, y) = \left[I_w - S_{padd-x} - S_w + \left(\frac{S_w - Cross_w}{2}\right) + 2 * C_w\right], \left[I_h - S_{padd-y} - S_h + \left(\frac{S_h - Cross_h}{2}\right) + 2 * C_h\right] \tag{14}$$

$$Q^2_{center} = (x, y) = \left[I_w - S_{padd-x} - \left(\frac{S_w - Cross_w}{2}\right) - 2 * C_w\right], \left[I_h - S_{padd-y} - S_h + \left(\frac{S_h - Cross_h}{2}\right) + 2 * C_h\right] \tag{15}$$

$$Q^3_{center} = (x, y) = \left[I_w - S_{padd-x} - S_w + \left(\frac{S_w - Cross_w}{2}\right) + 2 * C_w\right], \left[I_h - S_{padd-y} - \left(\frac{S_h - Cross_h}{2}\right) - 2 * C_h\right] \tag{16}$$

$$Q^4_{center} = (x, y) = \left[I_w - S_{padd-x} - \left(\frac{S_w - Cross_w}{2}\right) - 2 * C_w\right], \left[I_h - S_{padd-y} - \left(\frac{S_h - Cross_h}{2}\right) - 2 * C_h\right] \quad (17)$$

Given a 64 digit binary number, $B$, in the following format,

$$B = B_{64}B_{63}...B_1B_0 \quad (18)$$

we assign the binary digits to the components $C^n_{ij}$ with the following mapping:

$$C^n_{ij} = B_{[16(n-1)+4(i-1)+j-1]} \quad (19)$$

where $1 \leq j \leq 4$, $1 \leq i \leq 4$, and $1 \leq n \leq 4$. The constraint that has to be satisfied with the quadrants are that the widths and heights of all 4 quadrants have to be divisible by 4, such that $Q^n_w\%4 = 0$ and $Q^n_h\%4 = 0$. Fig. S1 shows the skeletal components of the stamp and several variables used in Equations 1-19.

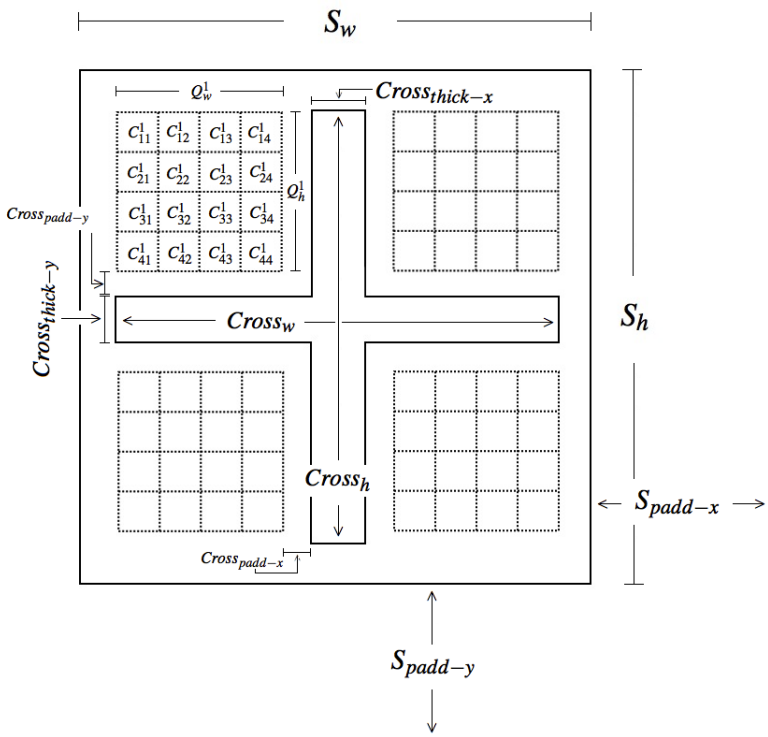


Figure S1: A blueprint composition of the stamp. The stamp is divided into 4 quadrants, each divided into 16 components to encode a 64 digit binary number. Each component encodes 1 out of the 64 digits and stores a 1 or 0 by using pixel colors. Bright components store a 1 and dark components store a 0.

*Step 7: Detecting OPCI stamped images*

Users surfing the web encountering our OPCI stamped images should be able to click on them to redirect to the respective product collection pages. In the case where the stamped images are not clickable (loss of link), do not link to a product collection page anymore, or link to a wrong page, the "Detector" plugin can fix all those issues. The detector plugin changes the "href" attribute of links around stamped images that lead to wrong links and builds from the OPCI binary stamp correct "<a>" link tags around stamped images.

As a user's browser loads images from different websites, the "Detector" plugin can be activated to scan each image for an encoded OPCI. Images that are not readable and tainted by cross-origin data security are sent to the proxy server for decoding, upon user request.

We use Equation (19) to reconstruct the binary number from the pixels on the stamp and convert the binary number to an OPCI decimal number.

Fig. S2 describes the flow of information and commands between a user's browser viewing images and additional proxy servers handling requests to decode OPCIs in stamps. Platforms such as Google images, Pinterest, Facebook and others can decode the OPCI from stamped images to build and correct their links. However, until such platforms start adopting this task, we made it simple for any user to detect these stamped images across the internet by using our "Detector" plugin. In fact, even on a platform such as Instagram that completely disables user links and prohibits the modification of its webpage code, the "Detector" plugin successfully re-builds correct product collection links around the stamped "hook" images allowing traffic to flow out of Instagram and into a user's product collection page. The legality of this matter is left to the user's discretion.

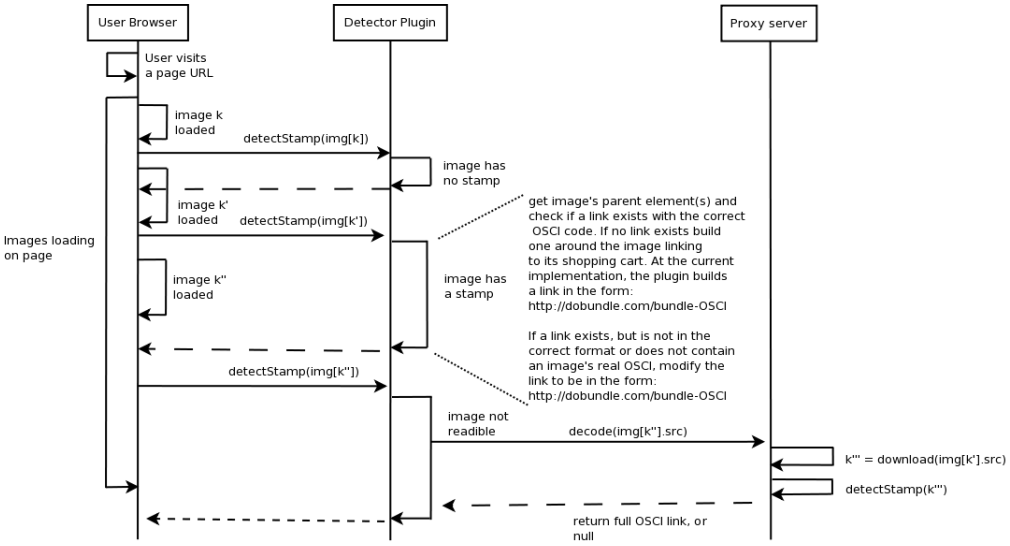


Figure S2: The flow of information and function calls among the browser, plugin and proxy server. The server is called only when the pixel data of an image cannot be accessed in a browser due to cross-origin security protocols.