

Tor-Like Anonymous Courier System For \$, Data, or Physical Goods Transport

PROBLEM:

A. Sender has an important possession, (Money, Information, A Toothpick, whatever) and he would like to prevent the recipient from finding out where the good came from, either because he doesn't trust the recipient, or because he believes the recipient and/or himself is under surveillance for some reason.

B. One could use a standard single or non-random trusted "route" of people to deliver the goods in question, but it only takes one weak link (a squealer or double-agent) to:

1. Confiscate and disseminate the item in transit, or perhaps worse,
2. Trace it back to you and/or trace forward (if that's even a term) to the recipient. Both might be equally bad, or one might be worse than the other, depending on the situation at hand. Regardless, one would have it in his best interest to prevent both if at all possible.
3. Enable agent provocateurs or double agents from interfering with or playing off sides, especially if the adversary has the time to perform long-term observation.

SOLUTION: Create a randomized and secure chain of couriers and Drop Zones (DZ) in which no individual in the transport chain knows anything more than his own instructions for picking up the item from one DZ and dropping it off at another (with the exception of the initiator, and potentially the recipient, albeit perhaps to a lesser degree). This basic procedure has been around for probably thousands of years in certain circles, but this is the only one (as far as I know) that employs a random, need-to-know-basis.

THE PROCESS

MATERIALS:

1. Graphing/scientific calculator or computer application that will do random integers within a range, data lists, etc.
2. A pen and paper or a computer spreadsheet application. Note that it should be secured in some way via encryption, steganography, etc.
3. A computer (PC or PDA) that can generate and/or utilize PGP/GPG crypto keys to encrypt small amounts of text data.

PROCEDURE:

1. Determine the method for random selection, i.e., should one use totally random data sets that might have multiple iterations of a number, or should you instead keep taking random samples till you get a data set that contains all unique integers? For example, is $n=\{1,3,4,3,7\}$ ok, or rather $n=\{3,4,2,1,7\}$? This could be dependent on the perceived attack model.¹

¹On one hand, a "truly random" set with doubles might eliminate permutation related attacks (?), but would be redundant. Basically, one must determine whether it's more secure to be entirely random, or whether going back to a DZ or person more than once would be too dangerous (IE, the person or place aroused the attacker, and thus is under surveillance).

MY THOUGHT: If there's only one or two "weak" attackers, "repeating" data sets would be OK, or even beneficial, assuming that the attacker(s) is not likely to look for the same person/in the same place more than once. For more sophisticated attacks where monitoring of some (but not all) people/locations is being done, repetition would do more harm than good, as an attacker could discover a DZ and/or courier and just stake him/it out and then seize the good and/or interrogate the transport agent.

2. Create one list of courier/DZ combinations, or do each separately (See 1. [?]) using one of the methods described above. When your list(s) is completed, figure out a distribution method. Should everyone get the entire list with the ability to only decipher their own public-key encrypted section of the whole, or should the “master list” be broken down into sections, such that each agent only has his or her piece of the instruction data? (This might be a security v. efficiency/simplicity choice rather than a fundamental security issue)

PERMUTED (NO DOUBLE ENTRIES) EXAMPLE

| <i>COURIER ID</i> | <i>DZ ID</i> |
|-------------------|--------------|
| 2 | 5 |
| 6 | 7 |
| 4 | 10 |
| 1 | 6 |
| 7 | 1 |

NON-PERMUTED (DOUBLES ALLOWED) EXAMPLE

| <i>COURIER ID</i> | <i>DZ ID</i> |
|-------------------|--------------|
| <u>2</u> | 3 |
| 5 | 2 |
| 7 | <u>7</u> |
| 2 | 3 |
| 1 | <u>7</u> |

In the first example, care is taken to make sure that random data sets are taken until a non-duplicated set is found. In the second example, the first **non-consecutive** data set is taken for both Courier and DZ ID lists, regardless of multiple iterations of a given integer. To make this more complicated (and perhaps more random), you could take any and all random sets, even if they were ALL the same number, but in most circumstances, this would probably defeat the purpose of doing all this in the first place. To do the same thing in a more effective way, one could incorporate random pick-up and drop-off times of goods at the involved DZs instead.

ROUTING DATA DISTRIBUTION:

There are a few different ways this could be done. Some may be better “all the time”, whereas some might be better in one set of circumstances than another. Is anyone an expert on this kind of information? What empirical data is already out there, if any? If none exists, testing of the theory in practice might be in order. Anyway, here are all the methods I can think of at the moment:

1. Distribute the entire list to everyone ahead of time. Each section is encrypted in order of receipt by the courier, based on the individual's public key. Thus, if Person 2, 6, and 4 receive the good in that order (based on the first table above), then Person 2, 6, and 4 will have their encrypted data on the sheet in that order (presumably from top to bottom). Pros- Redundancy.

Cons- Everyone has all data, making it potentially easier for an intruder or double agent to crack the entire chain.

2. The same as above, but only one copy of the entire routing data for the chain exists. Upon successful deposit of the item at the DZ, the courier rips off his encrypted routing info from the page and securely destroys it. The remainder is left with the item. Pros- Progressively minimizes method 1 weaknesses as the item goes farther down the chain. Additionally, only one trusted data distributor is needed, versus several partially trusted couriers and one fully trusted one. Cons- The first person still has all routing data and could pass it off to others, assuming he could crack the code(s). Also, a staked-out enemy or double agent could steal the remaining routing info if he was able to physically capture a known courier. This could be minimized by having separate locations for goods and routing info, but method 3 would probably make this complication unnecessary.
3. Each courier is given his own data separately. Either all are given the information all at once in the beginning, or it is done on an intermittent basis, either in a predetermined random order, or via a human agent that bases his alerts to the next member of the chain on secure check-ins from the last depositor. Here's where the random time element could come in, as opposed to in the actual routing instructions themselves (might be best when not knowingly under surveillance?). Pros- Eliminates time information from the encrypted instructions, making time-based attacks harder. Also, group could react to changing conditions better, in theory. Cons- Communications between "controller" and couriers might be unreliable, or even flat-out insecure. Security dictates that the fewer communications between agents the better.

DAMAGE CONTROL:

For maximum security, I would make an educated guess that Method 3 with randomized, pre-determined time intervals would be the best, if needed.² If complications arise, the group can compensate in the following ways:

1. Problem citing courier can take action by A) Calling off the operation and securing the transported good himself, or B) Contacting a Controller, who then contacts each remaining courier in the chain to call off the operation.
2. Courier secures good and does not alert anyone, including the Controller (if one exists) and the remaining couriers in the chain. Remaining courier(s) would then have to rely on a predetermined default pick-up attempt schedule, or regroup either in person or via some secure remote form of communication to figure out how to proceed in the future.
3. If item is better destroyed and given over to enemy hands, courier securely destroys good and securely contacts Controller/remaining couriers (or not, depending on Controller's existence and particular situation).

EXAMPLE CRYPTED COURIER INSTRUCTIONS:

Below is an example of what fields might be present in a given courier's encrypted routing instructions:

PICK_UP: L[x] (Where X is a # corresponding to memorized or securely stored location info)³

2 **THOUGHT:** Perhaps the following binary choice should be implemented based on urgency: If a good must be delivered quickly, time randomization should not be used. If time is not a problem, the following pre-determined scheme could be implemented: 1. Select a random time (0:00-23:59, probably in 1hr increments), T. If T is equal to or less than the current time, add T to the last drop-off time. For example, if Person X does a drop at DZx at 11:00, and the next random time is 4 (4:00), simply do 11+4 and set the next courier's pick-up time at DZx to 15:00. If the time goes past 23:00 (or 23:59 in 1 min increments), simply add the time as usual, but make it carry over to the next day. Ex. 22 (22:00) + 5 = 3:00 the next day.

3 Is it more secure to specify the recognizable location descriptor within the courier instruction code, or should it always be separate, using the same or a different key associated to the same individual? Distributed content v. having potentially more chances for the attacker to obtain and decrypt data. Would having 2 keys, one for instructions and one for location

DROP_OFF: L[y] (Where to drop off goods, based on rules for L[x] above)
PICK_UP_TIME: (A predetermined pick-up time, must not conflict with earlier drop-off time!)
DROP_OFF_TIME: (Beholden to same rules as directly above)
MISC_NOTES: (Extra info about things to look out for, special care to be given to the good, any intervals to check for non-received goods [if any], etc.)

NOTE: Fields 3-5 are optional, especially if no specific times for pick-up/drop-off are given.

Drop Zone Security, Explained

The following is excerpted from the publication *Spy & CounterSpy*, archived at <http://anonymity-portal.us/books/SpyCounterspy/fs015.html>.

How to set up and use a dead-letter box...

This article describes how deep-cover agents pass messages, documents, money, weapons, and other material between each other – without compromising their security. Neither agent knows the identity of the other. Nor do the authorities know what's going on.

The method described in this article has been used by foreign intelligence agencies and underground groups to thwart the counterintelligence and counterespionage sections of the FBI.

What is a DLB? DLB is an acronym for dead-letter box. It is also called a dead drop. A DLB is a physical location where material is covertly placed for another person to collect without direct contact between the parties.

Good locations for dead-letter boxes are nooks and crannies in public buildings, niches in brick walls, in and around public trash receptacles, in and around trees and shrubs, a third-party's mail box, between books in a public library, inside the paper towel dispenser of restaurant washrooms, and so on. The key to success is ingenuity. If the item being passed can be disguised as a discarded candy wrapper or hidden inside a cigarette butt, etc., so much the better.

DLB Protocol. The method described in this article was originally devised and perfected by the KGB for use in Britain and the USA during the cold war. But the technique is so effective it's still in use today – and is used by more than 30 intelligence agencies and underground groups worldwide.

When used by two people who have basic skills in countersurveillance, this method will confound an FBI surveillance team – as demonstrated by the FBI's inept handling of the cases involving Aldrich Ames, Jonathan Pollard, and John Walker Jr.

Tradecraft. You need to know three pieces of tradecraft to make this technique work.

Trick #1 – Pick a good site for your DLB. This means choosing a spot where you're *momentarily* hidden from view while you pass by (and either load or empty the box). It also means selecting a site that is easily accessible and in a public location.

Trick #2 – Use a separate set of sites to signal to your opposite number that you're ready to place something in the DLB, or retrieve something from the DLB.

Trick #3 – Use a foolproof signal that tells both parties that the material in the site has been picked up. This guarantees that the first agent can go back and recover the items if the second agent is unable to make the pickup for some reason.

Step 1: The *ready-to-fill* signal...

Let's suppose that you need to deliver a document to your contact. The first thing you do is transmit a "ready-to-fill" signal. You need to tell your contact that you're ready to fill the DLB with your material.

For example, you might place a piece of chewing gum on a lamp post at a pre-arranged location at a pre-arranged time (perhaps the second Tuesday of each month at 1:30 pm).

The trick is in using signals that can be easily seen by a lot of people. This means that your contact does not have to compromise his/her security while reading your signal.

Be sure to use a *ready-to-fill* signal that can be easily seen by a lot of people.

Step 2: The *ready-to-pickup* signal...

When your contact sees the *ready-to-fill* signal, he/she will send a *ready-to-pickup* signal. Again, this signal must be sent at a pre-arranged time and location, say at 2:00 pm. It might be a chalk-mark on a traffic signpost or back of a park bench.

When you see the *ready-to-pickup* acknowledgement, you must fill the DLB within 15 minutes (ie by 2:15 pm). After placing your materials in the DLB, you immediately return and remove your *ready-to-fill* signal, thereby indicating to your contact that the box is filled.

Don't fill the DLB until you see the *ready-to-pickup* acknowledgement.

Step 3: The *all-clear* signal...

Upon seeing that your *ready-to-fill* signal has been removed, your contact goes to the DLB and retrieves the material that you've placed there for him/her. This must be accomplished before a pre-arranged deadline, say 2:30 pm. Your contact then returns and removes his/her *ready-to-pickup* signal, indicating that the box has been emptied.

When you see this all-clear signal, you leave the area. However, if you don't see the signal by a pre-arranged time, you return to the DLB and retrieve the material in order to prevent it from falling into unauthorized hands.

This system of signals can be made even more secure by using positive acknowledgement signals instead of simply removing existing signals, of course.

When you see the *all-clear* signal, you can leave the area. If you don't see the signal, return to the DLB and remove the material.

Providing security for your DLB...

NOTE – The FBI does *not* want you to know this.

To maintain watertight security for your DLB, simply weave

Weave a number of *fake* DLBs into your routine on a

a number of *fake* DLB locations into your routine on a daily, weekly, or monthly basis. Narrow passageways between buildings, covered pathways in public parks, nearby dumpsters behind restaurants... all these are ideal.

Simply make it a point to walk past these fake DLBs *on a regular basis*. Remember, each DLB is located such that you'll be *momentarily hidden from view* as you pass it. If you're under surveillance, the goons will go ballistic. They'll need to place an agent at each suspected DLB *at the precise moment you walk by*.

If you've chosen your sites carefully, there's no other way for the goons to monitor these locations. If you have three or four fake DLBs that you regularly walk past, you'll soon notice the *telltale pattern of strangers* who just happen to be loitering nearby at the instant you're momentarily hidden from general view. When this happens, you've detected the presence of a surveillance team. Suspend your covert activities until the surveillance passes.

SURVIVAL TIP – Even if you're not using DLBs, it's a good idea to walk past fake dead-letter boxes as a part of your weekly routine. I've caught more FBI gumshoes than I can count with this one simple countersurveillance technique. To date the FBI trainers have been unable to develop a defense against this particular countersurveillance maneuver – and you just haven't *lived* until you've seen the facial expression of an FBI spook who suddenly realizes he's been *made* by the target of the surveillance operation.

daily, weekly, or
monthly basis.

COMMENTS AND FEEDBACK:

I welcome any and all comments and suggestions about this idea, especially if you've found any potential weaknesses or improvements that could be implemented. Even better, test this scheme or something like it out with a group of friends your self and see how well it works (or doesn't)! Please send all correspondence to Andrew at:

E-mail: firefox-gen@walala.org (Public keys are always nice)

Web: <http://anonymity-portal.us/>

AIM (With OTR GAIM Plug-in please!): PowerPenguinNIX*

*You can get OTR (or Off The Record) at <http://www.cypherpunks.ca/otr/>. Thanks.