Safe and Private Data Sharing with Turtle: Friends Team-Up and Beat the System (Transcript of Discussion)

Bogdan Popescu

Vrije Universiteit, The Netherlands

Today I will talk about a project which aims at designing a peer-to-peer network for safe and private data sharing. The motivation for this work is a development that threatens to shut down peer-to-peer file sharing networks, and that's a recent tactic by the recording industry to take legal action against peer-to-peer type networks. So first I want talk about the peer-to-peer file sharing phenomenon: in general, its origin, some of the positive social aspects of such a thing, and the tactical attacks that a peer-to-peer network is subject to. I will then focus on a specific attack that motivates our work, namely illegal users being sued, and discuss possible defences. Our solution, which we call Turtle because as you will see, it is slow but safe, cannot reach the performance of general existing file sharing networks, but at the same time we think it does a good job in protecting users against legal harassment.

Peer-to-peer file sharing is a phenomenon that started around 1999 with a company called Napster. It was really a place for facilitating people to directly exchange music in the form of mp3 files, and it was very popular from the beginning because people could get everything for free. At the same time it was also highly controversial because most of this music was actually copyrighted, so the recording industry perceived this as a major threat, and subjected peer-to-peer networks from the very beginning to multiple attacks. Before moving on to that maybe I should answer the question, are peer-to-peer networks any good, should we even work on protecting something that is mostly used to infringe copyrights? And actually I have some good to say of peer-to-peer networks, probably the most important property of such networks is that information cannot be censored. Basically once a piece of data is injected in such a network it takes a life of its own, and it stays there as long as there are people interested in that thing.

We use the example of protecting people sharing music because this is what mostly happens today, but it's easy to extend this to an example where people use a peer-to-peer network to, lets say, share independent political views, and I think it's worth the effort to work on protecting people if they suffer harassment.

So what types of attacks are there against a peer-to-peer network. The simplest attack is to go against the company for running a query service, and this is actually what brought down Napster. To counter this just move to a system such as Novello Kazaa, which has query processing, and this is what's going on today.

Another type of attack is to sue the company that writes the client software, and such an attack has been attempted against the developers of Kazaa in the

B. Christianson et al. (Eds.): Security Protocols 2004, LNCS 3957, pp. 221–230, 2006.

US, but again to counter this just move the company offshore, or underground, or whatever. Another form of attack is to attack the content and this basically means to inject bogus content into the network so that users have a hard time finding the actual relay node. There are also various attempts to do content tracing that involve counting how many users share different files, and it is also possible to counter this attack. Finally, the last attack in the arsenal is to sue illegal users, and so far more than a thousand people have been sued in the US for sharing files, and so far this is the biggest threat, and this is what we are trying to counter with our work.

Matt Blaze: You claim that's the biggest problem but, in some sense the first attacks that you mentioned seem to be more problematic, because they're attacks on an infrastructure based on the use that certain people perceive it to be put to, whereas the final attack of going against individual users is an attack on people for doing something that is at least in fact illegal. Those lawsuits may be very heavy handed, they may be misguided, and some of those prosecutions may be false, but at least they're exercising the legal system to deal with something that is in fact illegal. Couldn't we argue that attacks on the whole infrastructure based on one particular user are a bigger problem?

Reply: I see your point. The thing is, if you can sue people for sharing music. . . I mean, there are enough powerful lawyers that can go against anything – then if you share political views there are some pretty high placed people that don't like that, and they can go against you in the same way as people go against for you sharing music, it is the same model basically. There's always somebody powerful enough that doesn't like what you're doing, so if you have a system architecture that just lets powerful people attack you in this way, then they can shut down everything, I think.

Bruce Christianson: The argument is that a system designed so the simplest approach for the opponent is to attack individual users on an individual basis is flawed. That's the argument. The fact that the attacks on the individual users are legal attacks is beside the point.

Reply: Well, it's not flawed, it's what's happening at the moment.

Bruce Christianson: Well it's not optimal from the point of view of good system design.

Reply: Yes, we want to see if we can counter this, and OK, probably there's going to be a whole debate whether this is ethical or not.

Matt Blaze: The problem is that file sharing is illegal, right, and you argued that that's the problem, that the laws against file sharing shouldn't be as strict as they are. Then making it difficult to find the people who are doing it seems like a very indirect solution to the problem, the direct solution to the problem should be a change to the laws.

Frank Stajano: I believe that the point is this. Let's assume that this infrastructure, which is mostly exploited currently for file sharing here, is something

some people use for protection against censorship. Now this is a system that technically allows singling out the individual users who participate in that and beating them over the head, then it's not a good enough system to protect against censorship. I want a system where you can't find the heads.

Matt Blaze: Oh I understand that argument, and I agree that that's a desirable property for a system. I was just arguing in this particular example.

Reply: Another point we noticed is that, OK, maybe we want to change the law but that's very hard. Designing a system that allows us to protect against these laws is simpler.

Ross Anderson: Well there's a paper by George and me at the 3rd Annual Workshop on Economics and Information Security, next month[1]. To minimise the incentives for censorship, and also maximise the incentives for resistance, you should have one big system that in effect is a federation of fan clubs. So if an artist decides that it's stupid to go suing all their fans, then fine, all of his fans can share the music, they can love him, they can hopefully buy more of his albums. If they think suing all their fans is what life is about, then let them sue all their fans, and their fans stop buying. Eventually they'll figure out which works work better.

Reply: Yes, and this is good example, it's not artists that use their kindness, it's their record company. I don't think the artist has that much to say about it.

Ross Anderson: But if you have a federation of fan clubs, then the linkage to the artist is there, it's clear, it's definite, and if the record company is suing a particular artist's fans and as a result the fan clubs say, we hate that artist, we're not buying his records anymore, the artist himself is constantly hurt.

Chris Mitchell: It seems to me that the message of your talk, with some caveats, could be taken as there's people breaking the law out there, and they're being prosecuted, and we want to help them avoid being prosecuted. Now I realise that's not your message, but why are you presenting it as if it is your message, using words like attack when you mean a company seeking legal redress for breaches of copyright. Why not say, my objective is to enable peer-to-peer sharing for those people who wish to do it for legal reasons, rather than to stop those people who wish to prevent it for copyright reasons. Why not present the same talk but take out all the stuff designed to annoy record companies?

Reply: Well, it is fun to annoy record companies.

Chris Mitchell: But some of us believe in the rule of law.

Bruce Christianson: Then it is also legitimate to consider the next challenge that may be faced by law enforcement.

Chris Mitchell: We can enable the exchange of information that would otherwise be censored, and I think that's a very reasonable thing to be doing, but

[1] See George Danezis and Ross Anderson, 2005, The Economics of Resisting Censorship, IEEE Security and Privacy 3(1), 45-50.

why not present it as that, why are you presenting it as an attack on record companies?

Reply: This is what people can build, and this particular attack against the users would not work if such a thing were built.

Mike Roe: We're in danger of getting stuck in an argument about the ethics of whether we should be doing this or not, rather than discussing whether or not the protocol actually works. It's always the case that new protocols are presented here in an adversary form.

Reply: Because it's easier to present a talk in this way.

Bruce Christianson: Then let's apply a bit more abstraction for the sake of the argument.

Mike Roe: Without necessarily taking sides as to which of the two participants in this protocol we think should be gaining out of this.

Matt Blaze: Here, let me make everyone happy, if we can design a system that will allow file sharing in this very hostile legal climate that file sharing currently exists in, then it will surely work well for political dissidents.

Reply: Yes, exactly. So why is attacking the user such an effective mechanism? Well research has shown that most content in such a network is supplied by a small fraction of users, and this has led to the so-called "Crush the connectors" strategy from the RIAA, which is very simple: identify users sharing large numbers of files, reveal the content, from there log these transactions, and then they can see the patterns, and the net result of this is that exchanging content with strangers now becomes dangerous because you never know when a stranger is the adversary, so we get in trouble.

So to formalise the biggest threat model, basically you assume there's a peer-to-peer network in which a fraction of all the nodes are controlled by adversaries, and what we need to do is to prevent exposing the nodes, and this exposure can happen in two situations. The first is when a good node exchanges data with an enemy node, and the second threat is passive logging, when a passive adversary is capable of logging all the data exchanged between two good nodes. This corresponds to a situation when the Internet service provider would be obliged by law to log all the traffic exchanges certain people, or all traffic, that orients from a given node.

These are the only threats we're trying to counter, we are not particularly concerned about traffic analysis, since participating in a peer-to-peer network is not by itself a crime, it's only certain types of data exchange that may lead to legal trouble. And for the same reason we don't really care about strong views of anonymity. Before discussing what we propose, I want to talk a little bit about what most people think is the solution to this problem, which is anonymous file sharing. There are a number of systems that have been designed for such purposes, probably the best example is Freenet. The way they work is to just make it impossible to identify the source and the destination of a given data

exchange, and this is accomplished using primitives derived from early work on mixed nets and onion routing. And although in theory the adversary has no way to see you with such a system because he can't identify the source or the destination, in practice, as I will show next, this is false.

With an anonymous file sharing system, before reaching its destination data is routed to another peer-to-peer intermediate node, and the property here is that, in this way the source stays hidden. When a node wants to retrieve a piece of data shared by that source, it doesn't stop the source until the last relay.

Now let's see what happens when an adversary tries to retrieve a file shared by the source. First he will go to the last relay, and under our threat model the last relay is exposed, the adversary can just log the IP address of this last relay, and then take it to court for supplying something that's against the law, or undesirable, whatever. This last relay can get himself off the hook by proving somehow that it was just relayed data, but that most likely will involve exposing the previous node in the relay, and that's not desirable. What's really important is that these endnodes can be subject to enough harassment to make it unappealing for them to participate in such a network, and then you end up with very few nodes that are actually linked to a device in a network, and this is why this probably would not work. The important point here is that this exposure has not happened because any of the good nodes have not followed the proper security practices, but is more an intrinsic weakness of the system because it has this open service model. Basically anybody, including the bad guys, can access any piece of data, and this is why we think this may not work.

I will talk now about our solution, Turtle. The idea is rather simple, just make the peer-to-peer overlay based on social links. Two Turtle nodes connect to each other, if there is a social link between their users. Assume that they are friends in real life and they trust each other. And the communication between any two nodes is encrypted, and given the fact that two nodes that connect to each other in Turtle, are rather like people that know each other in real life, they can release encryption keys which fits very well with this peer-to-peer data line.

It's enough to have only one user to start such a network, one user who runs the Turtle node. He invites his friends to the network, and then they're secure between their nodes. This is what happens when a friend's friends join the network, and then friends' friends, and so on. A query is routed hop-by-hop, every node that receives a query broadcasts this query to all his friends on the link where it came from. Query results are routed differently than in the existing network systems, ours have to go back hop-by-hop, because only the friend nodes can be used, so they go back hop-by-hop, on a reverse route.

Bob Mayo: Are you assuming that I would trust someone that is ten hops away as much as the someone who is one hop away?

Reply: It doesn't matter because you never talk to somebody several hops away. You only ever talk with someone you know, you are telling something to your friend and your friend tells that to his friend, you don't deal with his friends,

that's the idea. So hits are passed back to the query originator who selects one hit, and then the solution is a virtual circuit, and that's all.

Frank Stajano: You are basing the security on jumping out of the system, you're not just doing in cyberspace you're doing in the flesh. But what's to stop the adversary to put in a person, an informant, a fake file sharer in the flesh? At some point you have to bootstrap this friendship by having gone to a party together, and getting drunk, or something.

Reply: That's where you have to tread carefully now. It all depends on you. But if you are careful enough, if you select your friends carefully, then your chances to be caught are very small.

Fabio Massacci: The nodes at one end may be very careful in selecting their friend, but they have a very different idea of what is a good friend from you so you can be caught anyhow.

Reply: Well it doesn't matter, because if I select my friends carefully, it's only my friends that know what I'm querying, and what I request, it's only *my* friend that can log what I want.

Bruce Christianson: The argument that's at stake here is the containment of failure, the idea is that if you foul up by selecting the wrong people then you only put yourself at risk, you can't put at risk the friends of your friends. This is potentially anti-transferable.

Reply: Yes.

Andy Ozment: I think it's actually worse than contained failure in the sense that if a single node is compromised, then they can follow the links from that person's machine all the way through the network. If we assume rubber-hose analysis here, they're not too picky about evidence, they can actually just...

Bruce Christianson: You might be guilty so we're going to sack you now.

Andy Ozment: Further, the problem with this is, I may be friends with you because you and I have the same taste in music, but we may have very different political tastes. The problem is when friendships are formed, and links are made, based on one assessment of friendship criteria, but people then use those links based on a totally different assessment such that, the friend of my friend may hate my political views so much he's willing to report me, or report this activity within the network, because he just joined this network to get music not to hear dissident political views.

Bruce Christianson: Or he may share your political views but really hate your music.

Reply: I think everybody has a limited number of people who they trust reasonably well not to turn them in no matter what, unless they really do something nasty.

Matt Blaze: It sounds like this has the unfortunate property that this is useful for file sharing by people in reasonably democratic governments where there's some rule of law, because you can't be forced to simply point fingers at people with no personal knowledge of wrongdoing, but quite weak against a repressive government that can beat people up and intimidate them without the rule of law limiting what they can do.

Reply: You are correct now, this doesn't work in totally oppressive regimes there has to be a certain amount of respect for the law. But I think there are many cases like this, I mean there are other things which you'd want to share.

Mike Bond: I think there is a fundamental problem here with this architecture of sharing certain types of information. What if this system allows people to get in contact, and to get, in a relatively easy way, data from people they didn't know too well without having to make good friendship bonds. So I'm friends with my first friend, we can exchange music if we want to, I'm not friends with anyone who's got Star Trek episodes.

Reply: Yes, you just use the six degrees of separation argument.

Mike Bond: You need a way to find the guy who's got the Star Trek episodes without telling your friend on the way that you like Star Trek.

Andy Ozment: Your arguments are that we're safe because all of our friends trust each other in the same way, and that we can get everything because all of friends know somebody who likes something completely different. These arguments, the argument for connectedness, and the argument for security, run very much counter to each other.

Chris Mitchell: A slightly different way of looking at this is, if your objective is to make available news which would otherwise be censored, what's the advantage of this over posting the news on newsgroups, or as many newsgroups as you can find?

Reply: Well we can censor it. It hasn't happened but in theory, if people know that keeping something on their own machine can lead to nasty consequences they're not going to be motivated to do this.

Bruce Christianson: Suppose the model is not just news but comment.

Reply: Exactly.

Ross Anderson: There's a curious possible side effect of a network like this. The latest econometric research suggests that there is no net cost to record companies for file sharing, they lose some sales but they make other sales. If I decide that I want to sample Northumbrian pipe music, I go to a file sharing system, I get someone, and say, yes, that's really nice I'll buy some CDs.

So the arguments for the record companies is that it is if that their enforcement causes people to move to semi-closed systems like Turtle it will cost them money.

Bruce Christianson: Even Turtle can have a search algorithm that works by sending out a key word that's meaningful to Trekkie fans, but to no-one else, and a Trekkie will respond to it.

Reply: I was going to talk a bit about the query ID protocol. Basically every query has this 128 bit random query ID, and every node has to remember the query, and the nodes also have to remember from which other node the hit ID came from, and this is for being able to establish this virtual circuit. Whilst the query issuer has a route to the node that actually has the data, he establishes the virtual circuit step-by-step, without actually knowing more than the next link. So you only have to talk with your friend, you don't have to talk to anybody else.

George Danezis: You could optimise this, because after the tenth hop, it doesn't matter anymore to reveal who the node currently having the query is, because it could have been originated by anyone. So can you optimise after coming back to this exit node, or would that break the protocol?

Reply: Well that may lead to a problem, it exposes the last node.

George Danezis: But it doesn't matter because it could be anyone.

Reply: Yes, it can be anyone but imagine you are the last node and that you have this lawyer coming to your house, saying you have been sharing this data, and you have to come to court. You are not very likely to participate anymore. You can get yourself off the hook by revealing the people who relayed this, but you don't like the harassment, and you are not the first relay in the network.

So what are the security properties of Turtle? Most importantly, each user is his own trust group. Basically, each user decides who his friends are, and this has a number of advantages, the most important one being this local identity which we discussed earlier. If you select your friends carefully the chances to get into trouble are quite small. Another interesting property is that such a network could be immune to the secret agent attacks: with these the adversary just creates a very large number of peer-to-peer nodes, and injects them into the network. In this case the adversary would not only have to create bogus nodes but also create real world identities for them, and make them friends with real people, and that's clearly very hard. There is also good protection against denial of service attacks, when a user injects bogus content, or takes down this node the people he would most likely hurt would be his friends, so we believe that motivation for malicious random behaviour would be reduced in this system compared to existing peer-to-peer networks.

So now the biggest question, will this work? The first issue is connectivity, if we just follow the social net we'll get the same coverage as with typical open peer-to-peer networks.

The second question is, are people on-line enough to keep this social graph. Again that shouldn't be any problem because ADSL is becoming very widespread, and it's not uncommon for people to have their computer on-line all the time. The biggest question is, can people who have lots of friends keep the social

graph connected? The question is whether these connectors will be able to cope with having to relay a large fraction of all the data that's been exchanged. We don't know yet if this is going to work or not. Our idea is to probe this social graph, and then run some sort of simulation, basically see how we are starting to perform against a normal protocol such as Kazaa.

Frank Stajano: Can you explain more clearly why the fact that you're exchanging keys off line provides confined damage?

Reply: No, exchanging keys off-line makes no difference, what makes the difference is the fact that you always establish connections to people you trust. The point of exchanging keys off-line is that you don't need a key server, it's totally decentralised.

Frank Stajano: If someone is targeted by the RIA lawyers, what stops these lawyers from saying under subpoena tell us all the people you have linked with, and so on recursively.

Reply: Well, I don't know the law very well, but I think they would have to prove that you have done something illegal to pursue a subpoena for something like this.

Bruno Crispo: But the way in which the system is designed it's more difficult to penetrate the system rather than, for example, Freenet.

Ross Anderson: There are rate limits on our lawyers so courts can only handle so many subpoenas, assume that a lawyer can subpoena a thousand people a year and that's about it.

Bruce Christianson: The property that Turtle has, is that in the other systems it's reasonably straightforward for an agent provocateur to act as a distribution node and to compromise a very large part of the system, whereas here it's much harder.

Andy Ozment: Of course it also increases the value of penetration. Let's say that we have a group of dissidents and we keep our friendship links based entirely on dissidence, then penetrating the network is slightly more difficult, but vastly more valuable because here you have a chain of dissidents and you can just follow up the electronic evidence. Essentially you create a system where you have this electronic memory that we discussed earlier this morning[2], but you remove plausible deniability because you have this connection evidence on an individual's computer.

Bruce Christianson: Your argument is absolutely right if you have a one-issue network, if it's all political opinion, or all "The Darkness is a good band." But using the same network for multiple purposes undercuts that analysis.

Reply: It's also what Matt was saying. If you're in a totally evil regime, then just the fact that you are sharing encrypted stuff with somebody else will lead to trouble.

[2] Bohm et al., *Controlling Who Tracks Me*, these proceedings.

Bruno Crispo: Yes, it's really for dissident networks because essentially you do a new key distribution every time you use the network. My neighbour doesn't have the key to my data.

Ross Anderson: This model could be combined with sufficient node-level deniability. Bruno asks me for a CD by an unpopular band, and I relay the request unknowingly to Frank who provides it, and somebody comes along and says, this Italian song is pornographic, and I say, start again, I don't speak Italian. If that provides me with enough deniability...

George Danezis: Lots of people have mentioned that this might work in soft places, and not hard places. The main problem would be that by traffic analysis you would be able to tell who is participating, but if participating itself is not incriminating because lots of people use it to share lots of different files...

Bruce Christianson: Yes, that's the assumption.

George Danezis: Then it will be truly difficult to penetrate, because beating people up is actually really expensive. If you have a small degree graph, and let's say it takes one day for someone to be beaten up, then it is one day for the very few people round it to go underground, or to start burning their hard disks, or whatever, so it is extremely difficult for the adversary.

Bruce Christianson: But the crucial point is the one made earlier, so long as most of the traffic is for one particular issue, it's much, much harder for the attacker to isolate individuals.

Stephen Murdoch: If one of your friends reported you to the adversary, whoever they are, can you find out who that person was so that you can maybe warn your other friends or harm the reputation of your ex-friend.

Reply: In a reasonably democratic society, if your friends rat on you, a few probably have to go to court and testify against you and then you'll see who testifies against you. But there's no electronic way to detect that.

Bob Mayo: The model for this is the need to establish trust with people. If you violate their trust, you're known, and you will be found at the bottom of the river or something. Presumably not just over one song. [Laughter]

Bruce Christianson: But maybe for a really bad album!