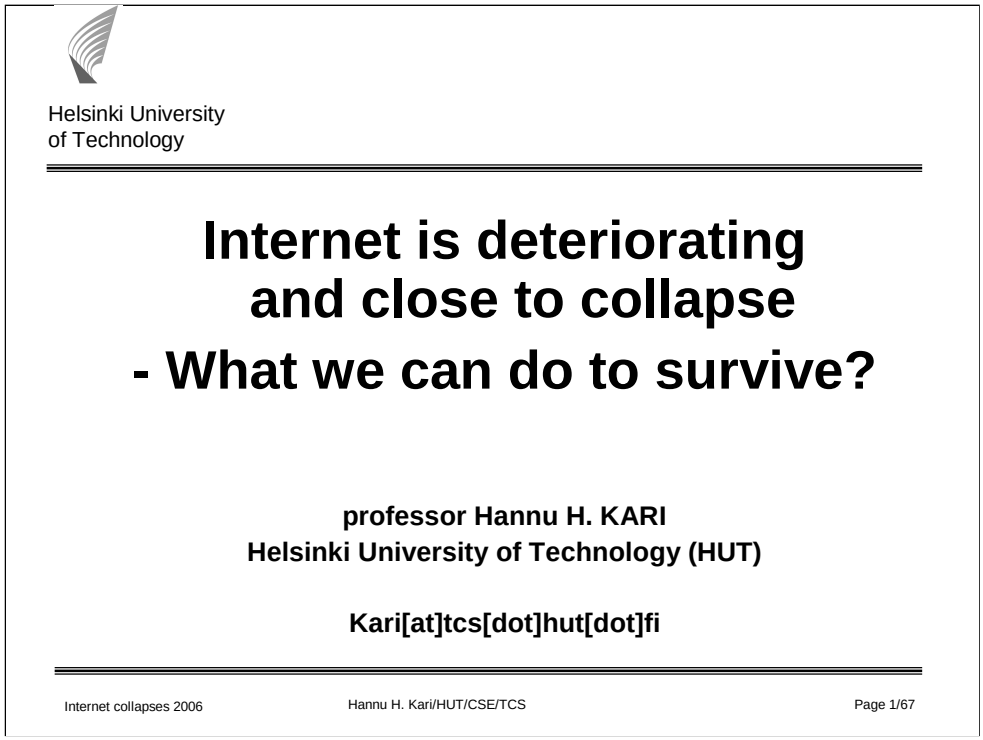


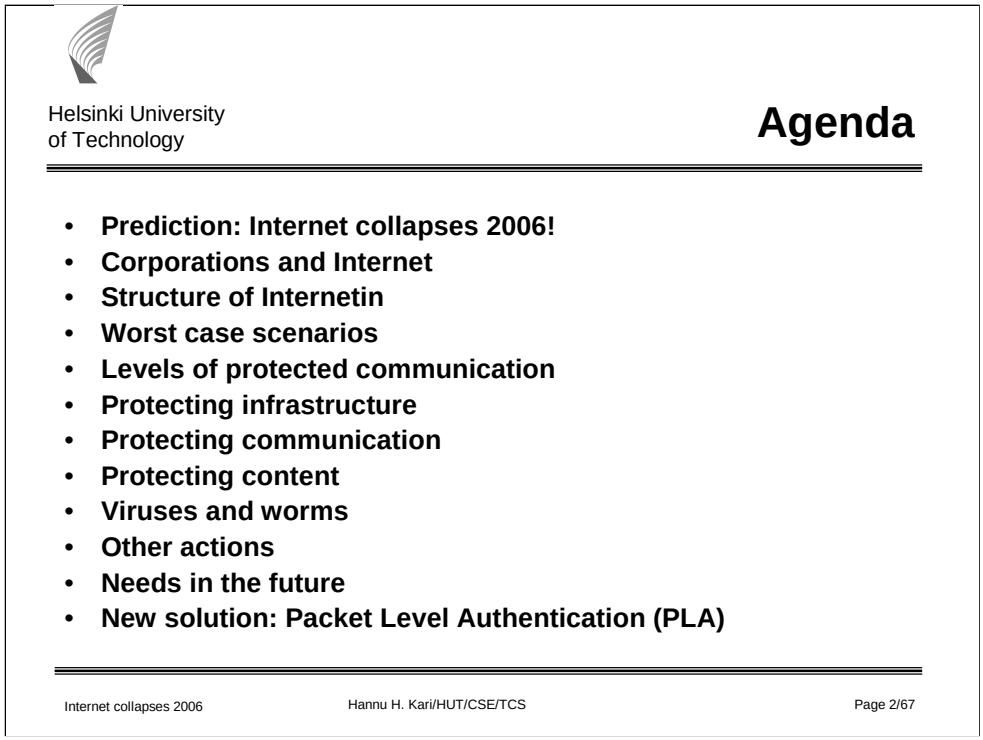
This slide set is prepared because so many person has asked more justifications and explanations what I really meant when stating: “Internet is collapsing by 2006”.

[This is a quick set up of slides, written in hurry, so please excuse my typos ...]

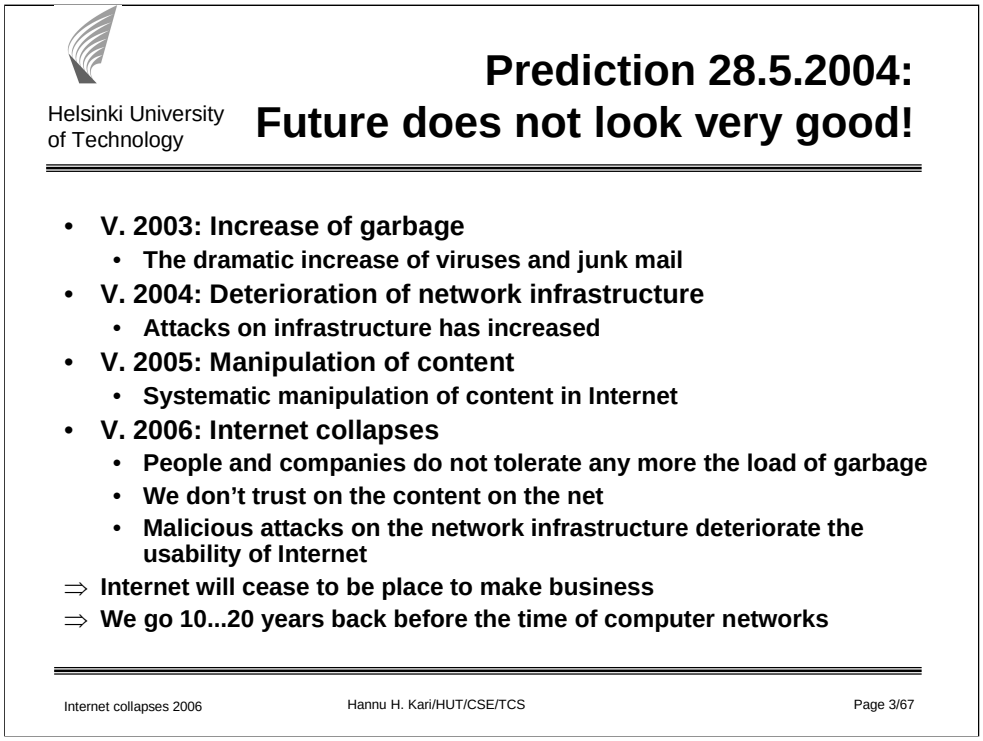
What I originally said was: “Due to viruses, junk mail, deterioration of Internet infrastructure, hostile attacks on Internet, and manipulation of content, all these problems will lead into a situation where companies and individuals will be totally fed up with the garbage and problems of Internet that they will cease to use it”. If we are forced to stop using Internet due to malicious entities (hackers, professional criminals, net-terrorists, etc.), our life will change dramatically!

Internet is not “one big computer that can be switched off”, but Internet is a way of thinking (“everything is on the net and always accessible”). Unfortunately, network is not always operating well, especially, due to hostile denial-of-service attacks, neither the data what we get is always correct. We can’t let children to go to Internet, since all the pornography, etc. that pops up to their faces...

Think about this: “Google gives us nice way to search information. What if someone wants to manipulate us by putting first those links that are more favourable for their interests. Information is almost correct, but we can’t detect the flaw!”



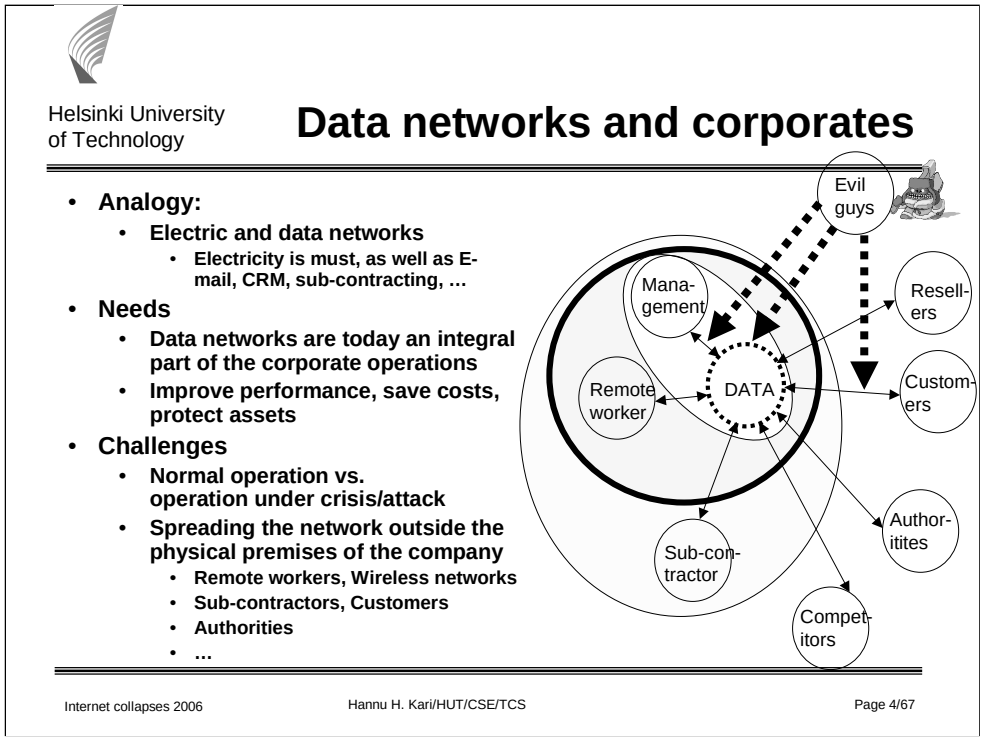
This is a short agenda for my presentation...



Virus attacks are the everyday problem. Research director Mikko Hyppönen, F-Secure, has bad examples of problems what viruses has caused: Internet network of a nuclear power plant was infected by a virus in Ohio, Hospital in Sweden was forced to move patients to another hospital, because their X-ray machines didn't work due to a virus, Australian railway system was shut down, airlines have had problems,electric power shortages , emergency phones have not worked, banking has had problems, etc. Nice link of Mikko's presentation is at www.mimesweeper.fi/pdf/Hypponen.pdf (sorry in Finnish). Mikko's presentation shows also how badly viruses (and also other attack) can impact our infrastructure, not only Internet itself, but also other critical infrastructures, such as electric power distribution, banking, airlines, railways, hospitals. And these virus attacks are not targetted against those infrastructures!

Additionally, we are facing more and more problems in integrity problems in Internet. How a normal user can verify that the email that she gets or web page or a document that she has downloaded is authentic? Majority of the junk mails come with forged sender information. It is possible to manipulate information of a corporate web pages without breaking into their servers (e.g., redirecting queries to an other server by manipulating/attacking DNS-servers, or modifying information at web-proxies).

All these problems in Internet deteriorate people trust on Internet. Finally, people are fed up with problems and stop using Internet any more...

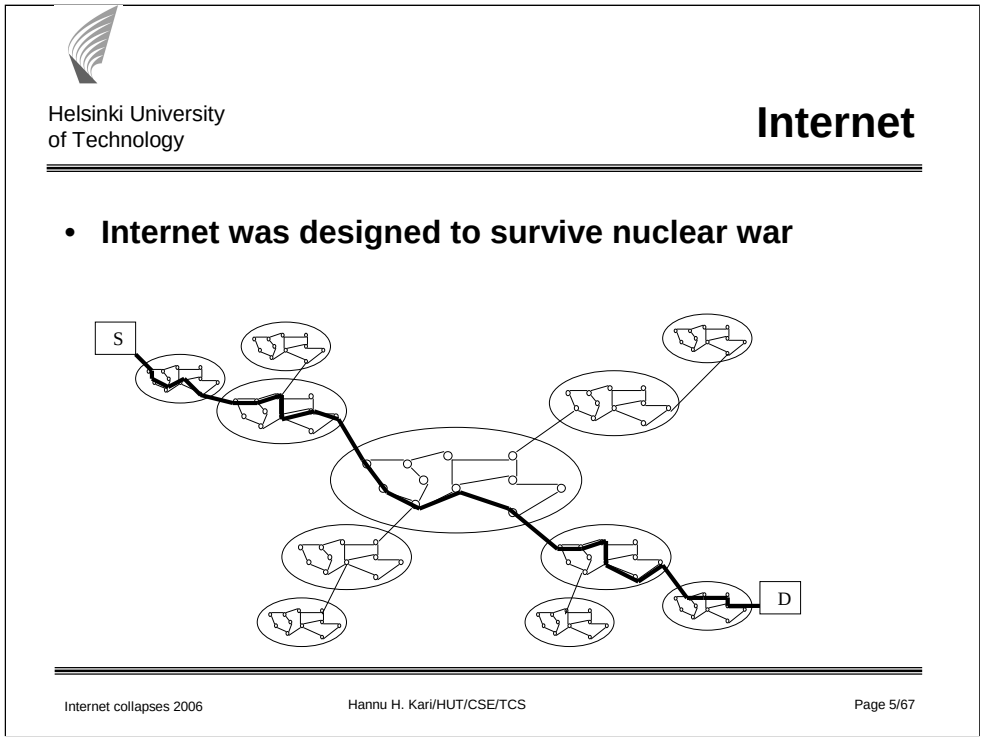


When companies used yellow pages (of the telephone catalogue) in adverticing their products, the system was operational all the time. Nowadays, companies are not only selling their products via web but also manufacturing and designing their products using computer networks. Corporations must share information amongst each other in order to do the business. But, the information must be controlled, who gets it, it must be correct, and it should come when needed.

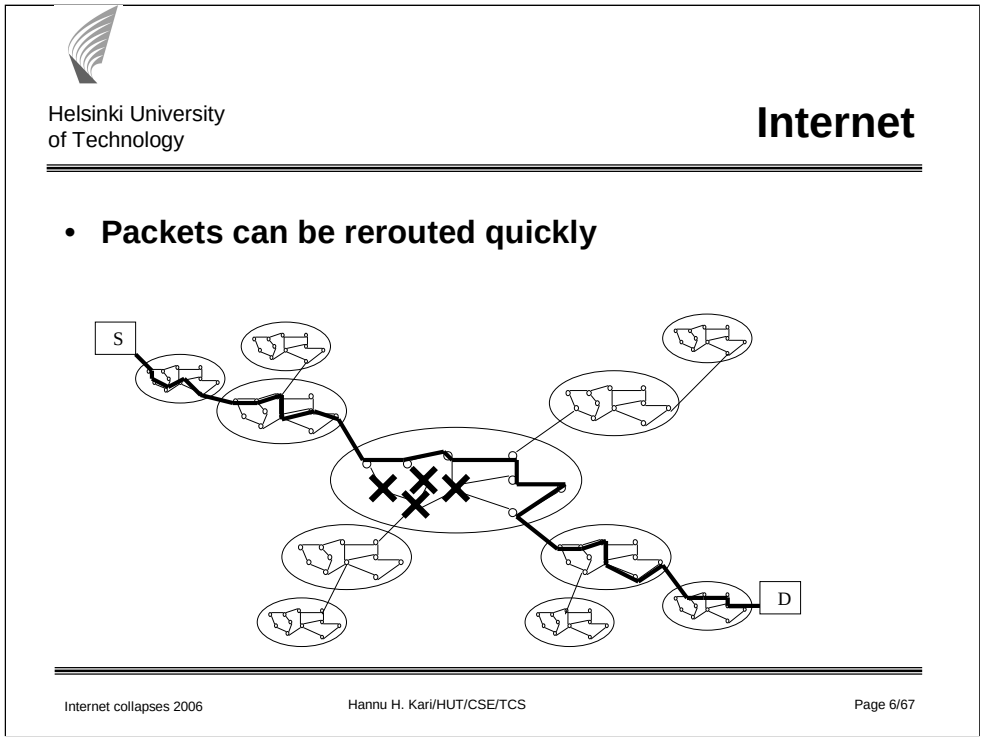
Possible attacks against the information are, for example, breach of information, manipulation of the data, but more and more serious problem is interruption of the data delivery (by paralyzing the communication).

Thus, corporations are as dependant on the communication as they are on electricity. This is because they must be increase productivity and be cost-efficient. Companies must react as quickly as possible in order to be in business.

Because companies must be connected to Internet, they are investing money on firewalls, virus protection, intrusion detection, etc. But all those measures are not enough, since the attacker can easily paralyze the comminucation with simple denial-of-service attack, that either jams the network connection or cuts the link totally. By using alternative connections to Internet, we can slightly improve our situation. Then, the attacker needs twice as many hijacked computers to do the same attack (in practice with zero additional cost).

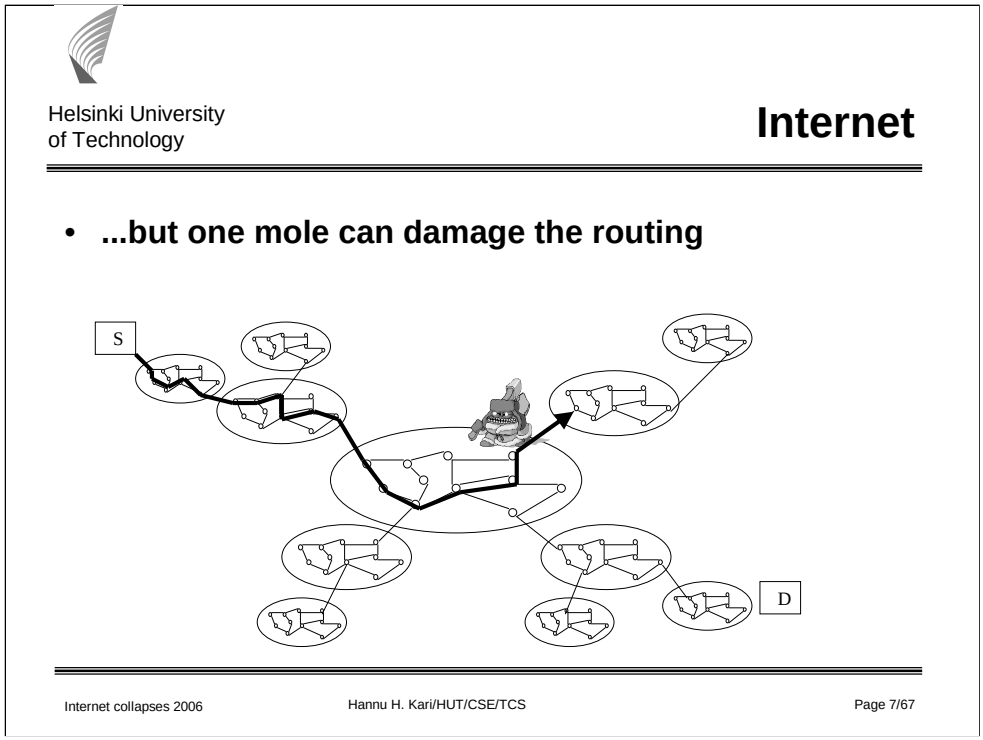


Originally, Internet was designed to survice nuclear war. Its one of the key design criteria was to be capable of routing packets from source (S) to destination (D) even in the case when large portion of the network is destroyed.



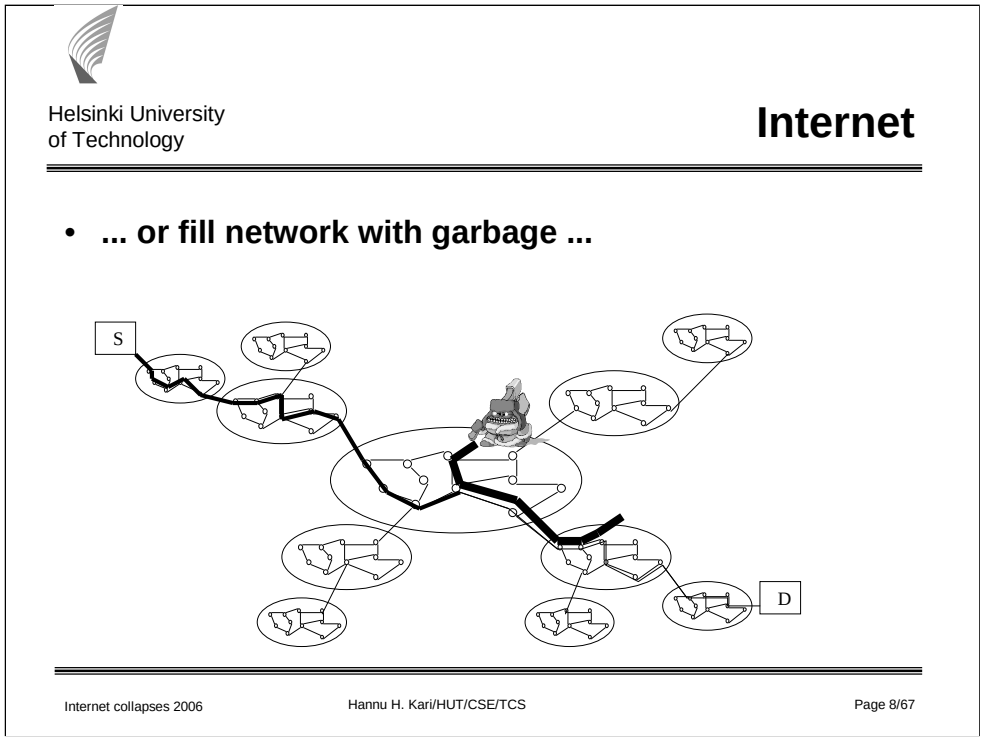
The routing is done per packet, and once the old route from S to D fails, the network will reconfigure itself and the following packets can be routed successfully to the destination.

This principle works fine in Internet, as long as all the nodes in the network are beneficent. Once a node becomes malicious, it may do serious harm to the system from inside (but that was never ever thought during the original desing phase of Internet!).



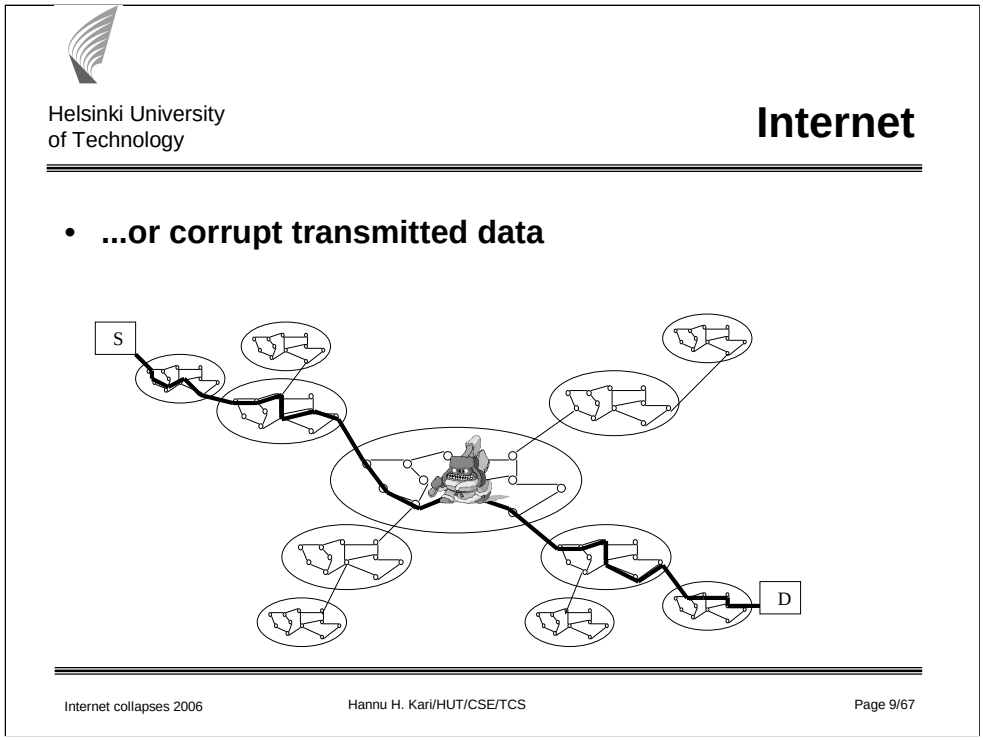
A simple attack to the system is to lie to other routers ”I have the best route to the destination, please forward all packets to me”. Since the network is designed to believe its neighbors, they will obey the new routing rule.

Of course, by adding some security methods, we can limit this kinds of problems.



A second attack to the network is to send large amount of garbage towards the destination and this way paralyze the real useful communication. This is a typical attack nowadays. More sophisticated attacks can be made against firewalls and/or server by opening large number of TCP sessions or by using large number of attacking computers together against the same target. Since any node in the network can send packets to any destination in the network, this means that not only the good guys can do it, but also the bad guys.

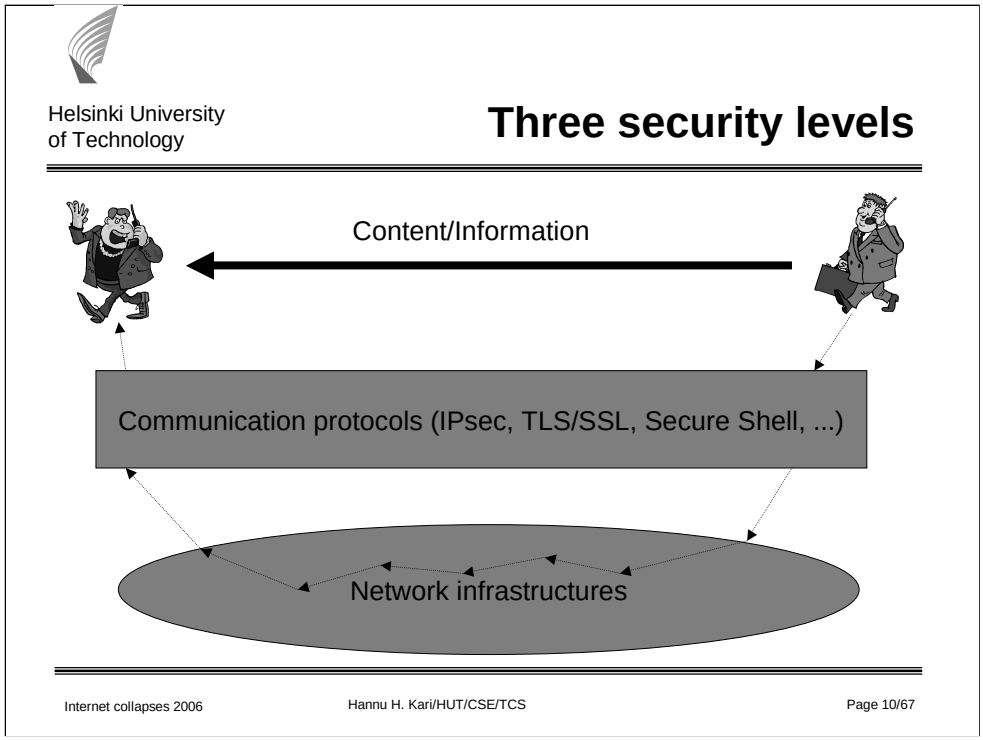
If the filtering of the imcoming traffic is done at the firewall of the corporate network (front of the web server), it does not help much, since the network interface is already jammed.



Typically, security solutions are end-to-end. For example, IPsec encrypts packets at the sender side and the receiver side then decrypts the packets. This means, that we can ensure the confidentiality of the data that is sent over public networks.

If the attacker modifies the packet in Internet (by randomly changing one bit in the packet), the destination node that does the decryption detects that the packet was modifies and discards it. This means that forged information can't be fed into the real data stream. However, network resources are wasted since the sender must resend the data again.

Sender and receiver has no means to detect at what location the data was corrupted since none of the intermediate routers is capable of verifying whether or not the packet's integrity is OK (this is because in IPsec the node needs to have security association established with the sender before the packet can be verified. Since IPsec uses symmetric keys in verification, then any node that has the verification key can also change the content and resign the packet).



We should think security on several levels

At content level, we must have mechanisms to verify that the content is correct. For example, is the mail authentic, is the web page created by the right company, is the sales data Excel-file up to date?

1) Communication protocols ensure that data that is sent over the public network is protected against eavesdropping, corruption, replay, etc.
2) But the most important is that communication is possible between the sender and the receiver. If the network infrastructure is incapable of delivering data, then we have no use of communication protocols and content integrity checkings, since we have no data packets.In critical organizations (such as emergency, military, police, etc.), we have very often private networks that are physically (or at leadt logically) separated from the public Internet. However, they are also vulnerable against attacks on network infrastructure, if the attacker gets an access to the physical media (cable, fiber, or radio link). Then, the logical separation (done using MPLS, ATM virtual channels, Ethernet's VLAN) actually gives powerful tool for an attacker to use existing traffic of another virtual networks in attacking one virtual network. This can be done easily e.g., by changing the unprotected MPLS lables or VLAN tags.Routers in the infrastructure will not detect this modification!

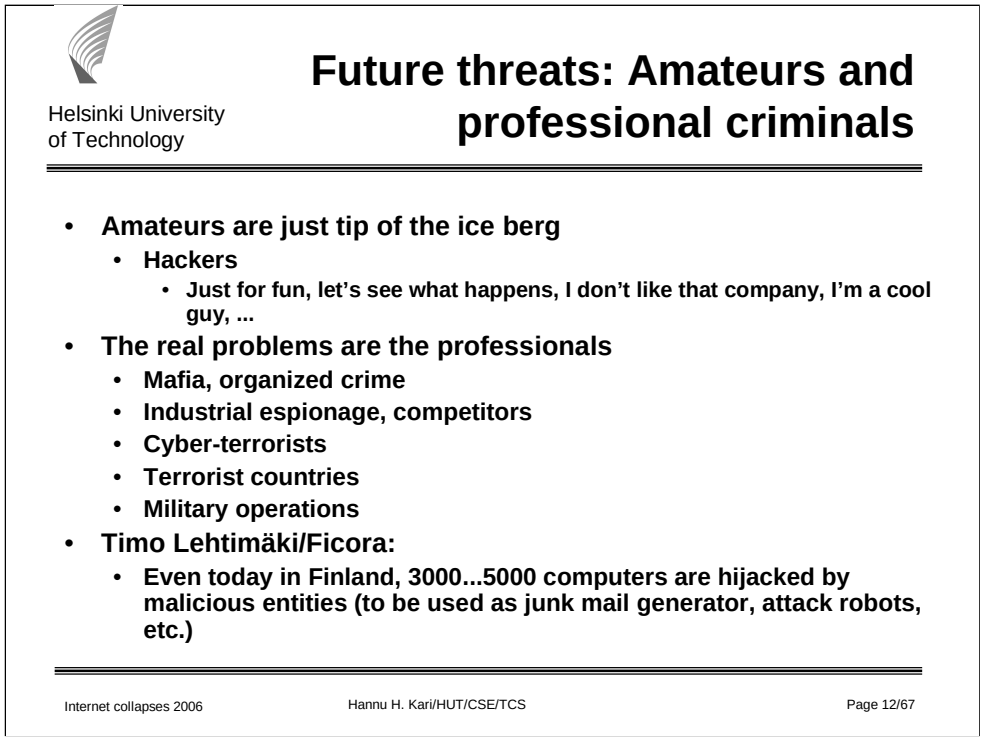
Thus, we must have all these three layers in good shape!

If Internet is not working, the impacts are severe on all sectors:

-Companies can't operate the normal ways, their business is either paralyzed or slowed down, their reputation can be seriously damaged, ...

-Most serious attack is on our financial system. If the communication to/from the banks is paralyzed, the entire society is in chaos. Since in many countries, Internet-banking (of individuals as well as small and large companies) is very common and even shops are using (VPN protected) credit card validation system over Internet, we can't anymore disconnect banks from Internet without also diabling large number of banking operations. Thus, if the Internet is unoperational, significant of monetary transmission is impacted.

-When people are not capable of paying their bills, buying groceries, of gas for their cars, they will become extremely restless. This causes serious impacts on the stability of the entire society.



Who would do attacks?

-The most visible attackers as those who do not work professionally. They may be skilled, but their interests are mainly to ”have fun” or ”to show up”. Their main interest is not to do serious damage in large scale.

-Absolutely the most serious problem is the professionals, whose main interest is to make money (or gain other interests over the attacked target). These guys know what they are doing, they have enough resources, they can basically do anything what we can imagine (and even more). Very seldom we hear about these guys since it is the interest of them and their victims to keep the incidences secret. For example, a big bank would not like to reveal that it is paying to organized crime some ”protection fees” in order not to be attacked. If that information were known by the customers, they would immediately change the bank.

Timo Lehtimäki, form Finnish Communications Regulatory Authority (Ficora), stated in one meeting that around 3000...5000 computers in Finland are infected by serious viruses or other malware at any moment. Once some of those computers are cleaned, there will be new ones infected. So, the balance remains about the same all the time. If these computers are used for attacking a certain target (such as a bank), they can generate distributed attack stream of 1...2 Gbit/s that easily paralyzes any normal Internet interface.

What kind of damages can be caused by Internet malfunction:

Direct damages are obvious:

-We loose money, time, information due to attacks

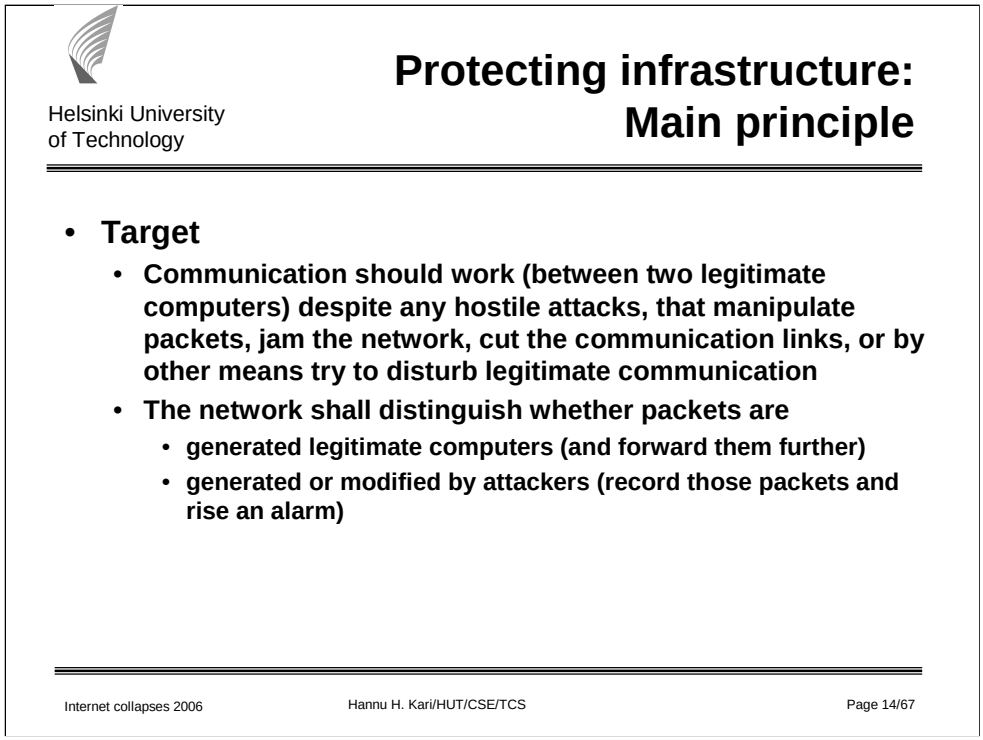
-We can't do our business any more

-Our reputation can be seriously (or permanently) damaged

Indirectly:

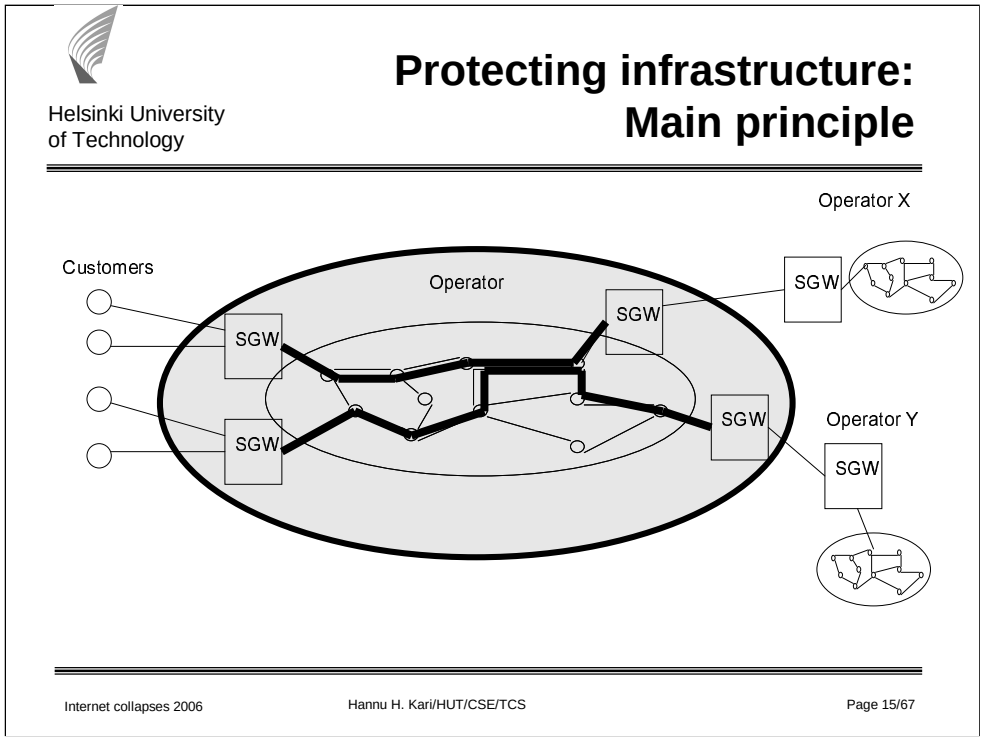
-Our society is not any more stable, riots, ...

-Especially in northern hemishpere (during the winter) it may be matter of surviving during cold weather. When there is –20 centigrade temperature outside and electricity fails due to attacks on infrastructure, the entire society is in danger.



If we want to have our network infrastructure operational in all situations, we must be capable of detecting whether the packets are coming from good guys or bad guys.

Since the attacker may have infiltrated into our internal system (or physically attached out transmission media), we must use cryptograhical tools to ensure the authenticity of the packets. We can't just rely on physical security.

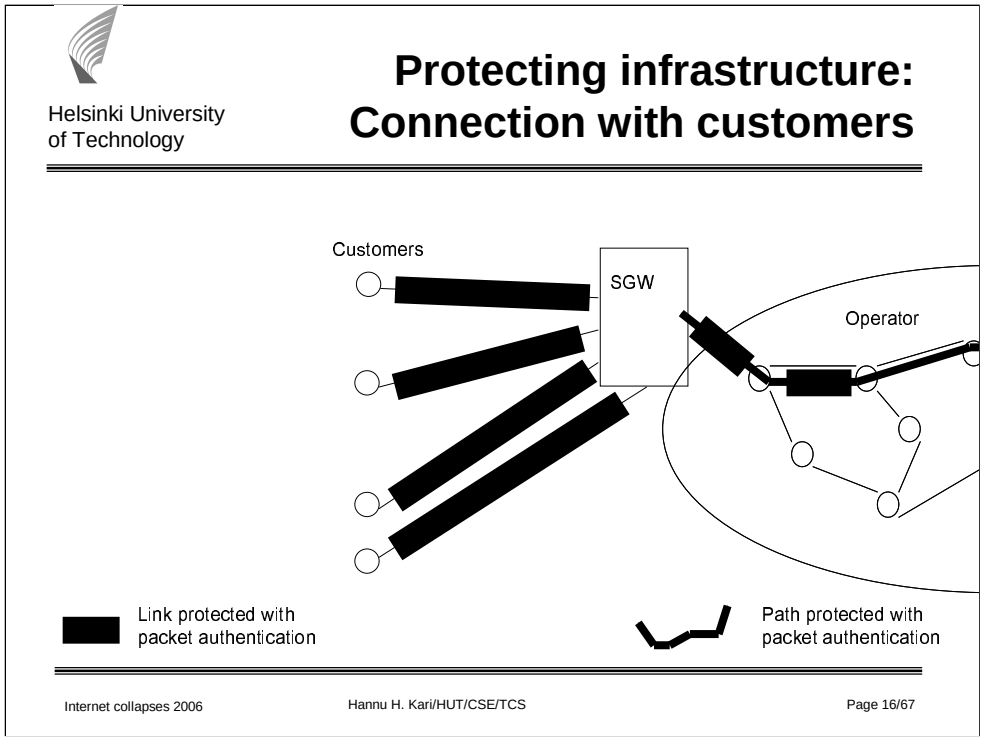


The infrastructure protection starts from the operators.

When an operator want's to be really sure that it's infrastructure is not under attack, it must protect its network on the borders but also "against attacks from the underground". Operator shall use security gateways (SGW) when communicating with other operators and its cutomers in order to ensure that its talks only with legitimate partners.

In addition, the operator must protect its communication links properly. Since links are very often long and unprotected (digged into ground), it is possible for an ttacker to get an access to the cable and install its own device into the system. If the operator has no security solution to protect the data (but trust on MPLS/ATM/VLAN level of data separation), it is very simple for an attacker to mannipulate traffic flows and cause serious damages for the traffic and also the reputation of the operator.

Thus in order to avoid legal and financial problems (as described in page 19), each operator must ensure that its network only carries legitimate packets.



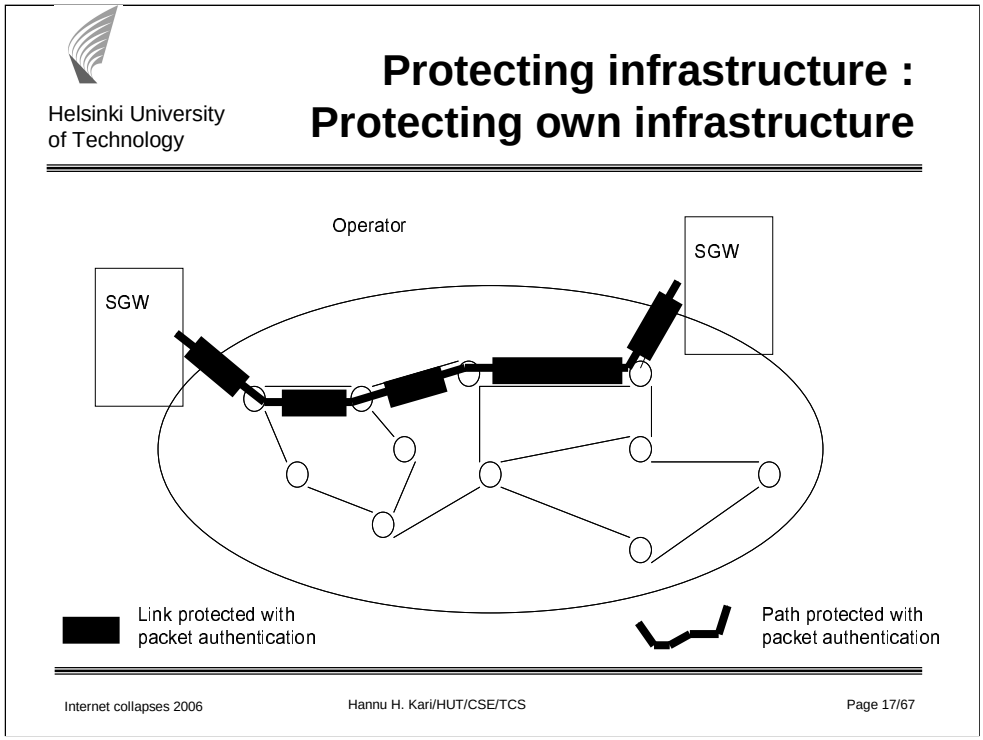
Operators must be sure whose packets it is forwarding. This means that operator must authenticate all its users and check that its users are sending only good packets. This requires some sort of authentication of the user/subscriber at the connection time and per packet level. Also, the operator must monitor the traffic and verify that user is following the rules (e.g., user does not forge sender IP addresses, send email with forged sender information, attack against other computers with flooding).

In case the customer behaves maliciously, the operator should immediately take preventive actions:

-cut down the communication (or dramatically decrease the capacity)

-notify the subscriber

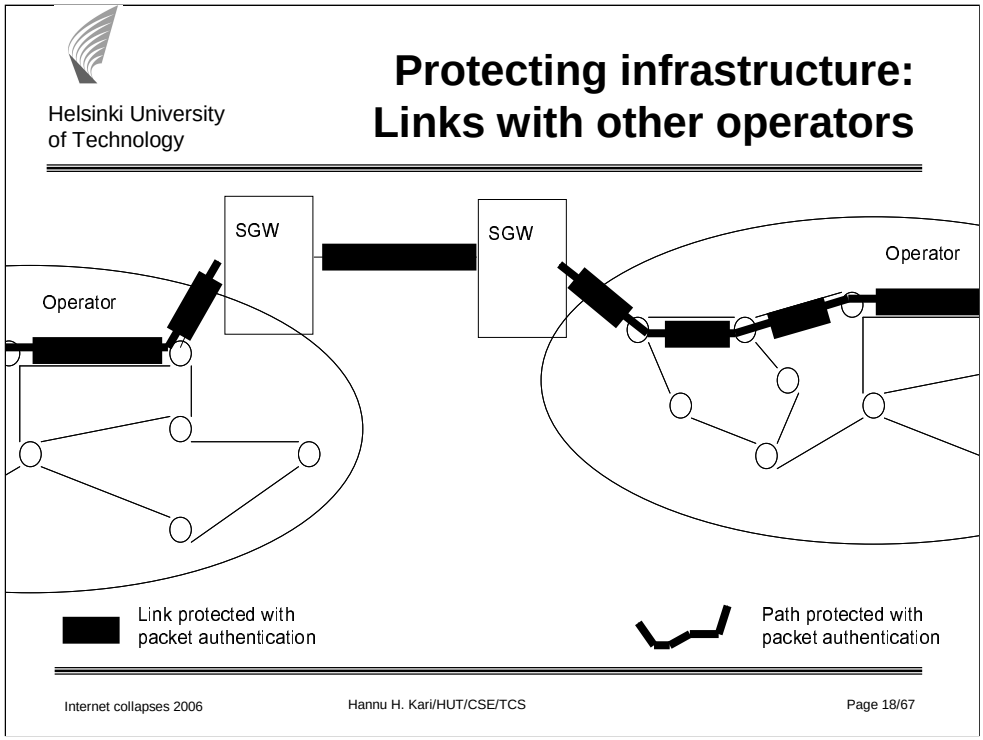
If the customer's computer is infected by a virus, it is better for both sides (customer and operator) that the virus is detected as soon as possible and its damages are restricted.



Like described earlier, page 15, the operator should protect strongly its own infrastructure.

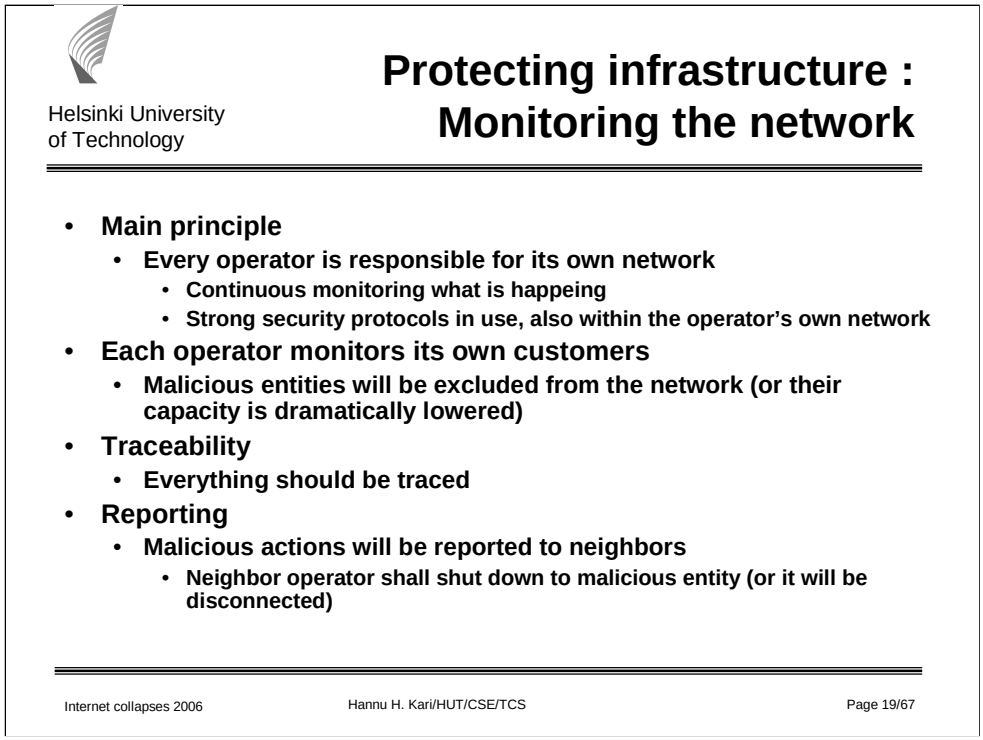
The main idea of link protection is to detect potential physical link attacks as the integrity of packets is ensures in every link. This means that an attacker, that has got an access to the physical media and manipulated traffice, will be detected already at the next router since it sees that integrity of the packets from the previous router is broken.

The main idea of the path level protection is to ensure that we can detect nodes that are compromised (e.g., the attacker has gained control of one of ther routers in the operator network). Then, the path level protection minimizes the number of nodes that each SGW needs to trust. In practice, SGW needs to trust only other SGWs.



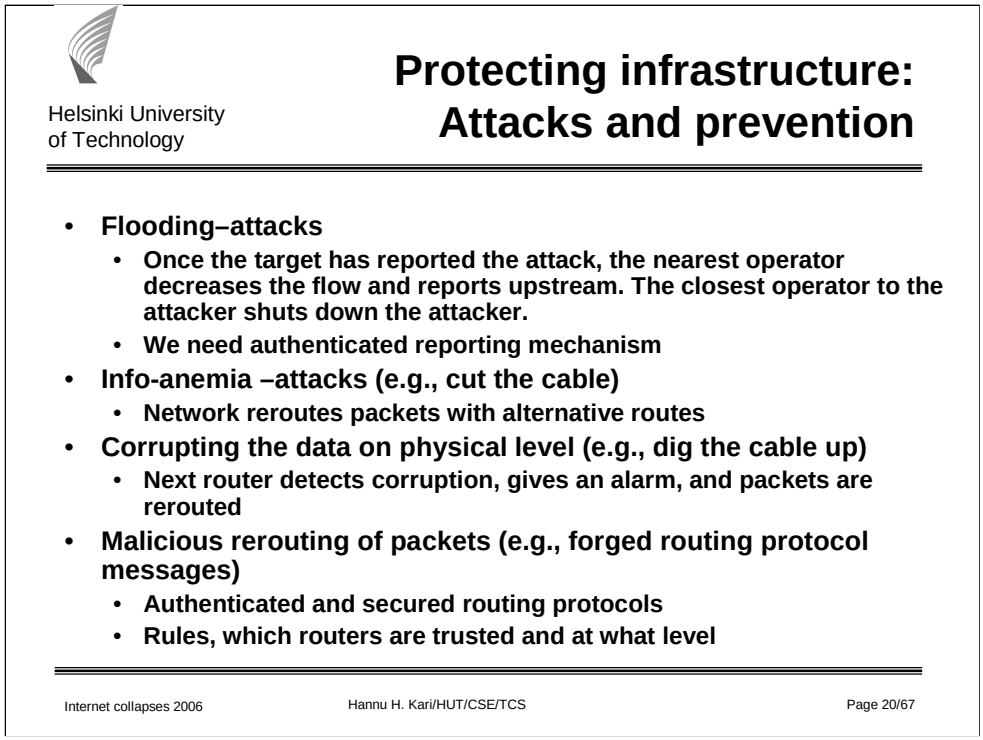
Links between two operators must be also protected. This is a mutual interest of both operators, since they want to be sure that no outsider is capable of injecting malicious packets to the link without their knowledge.

The solution is the same as within the operator. We can use either link protection or path protection depending on the physical implementation. Actually, from the SGWs' point of view, the solution is the same since their are only relying on the other SGW's, not the intermediate nodes.



So, each operator needs to monitor more closely what is going on in its network.. If some entity is behaving maliciously, it will be detected promptly and its damages are minimized.

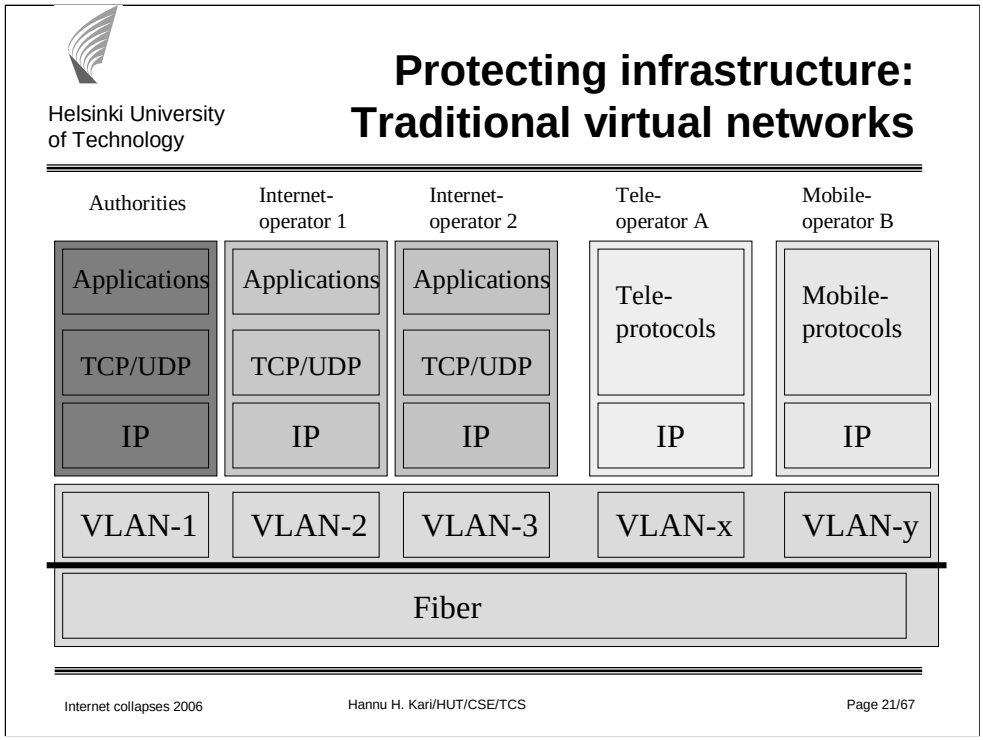
Since the tracing is done in real time, we don't need to keep log files for long time. The idea is not to store all log information for next three years, like some EU legistlations are proposing. Instead, the traceability means that once the attack is happening, it is reported immediately towards the upstream and each operator on the path should take protective actions to minimize the damages. If the operator does not act promptly to prevent/mininize reported attacks, it will be excluded from the community.



The best way to prevent the flooding attack is to do it as close to the attacker as possible. In order this to work properly, we need authenticated reporting mechanism, so that this reporting mechanism can't be misused. A good solution for this is to use Packet Level Authentication (see second half of this presentation).

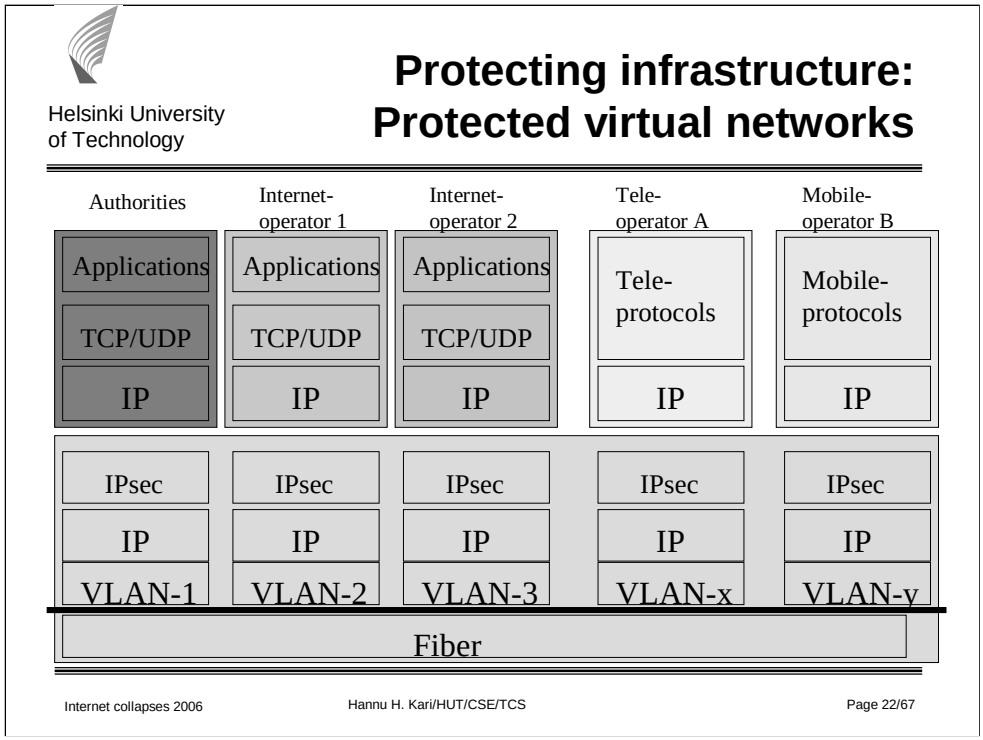
There is no way that we can prevent some one to cut our cable or corrupt our data on the physical media. But we can make the detection happening already at the next hop router (not the final destination of the packet).

Obviously, we need to have our routing protocols protected well enough so that our routers can trust when some rerouting is really needed.



Typically, many organizations are using various virtual network technologies (such as Ethernet's VLAN, MPLS, ATM's virtual circuits) in protecting and separating various (virtually separated) networks. For example, the same optical fiber can be shared with Internet operators, authorities, teleoperators, etc., by using Ethernet's VLAN tags to distinguish into which virtual network each packet belongs to. From the user's point of view, this gives total separation and the attack from one virtual network against another is not possible.

Unfortunately, the attack is very easy for any entity that has an access to the physical media. Since the VLAN tags in Ethernet (like in MPLS and ATM) are not protected, the attacker can easily suffle lables and use the existing traffic as an attacking tool.

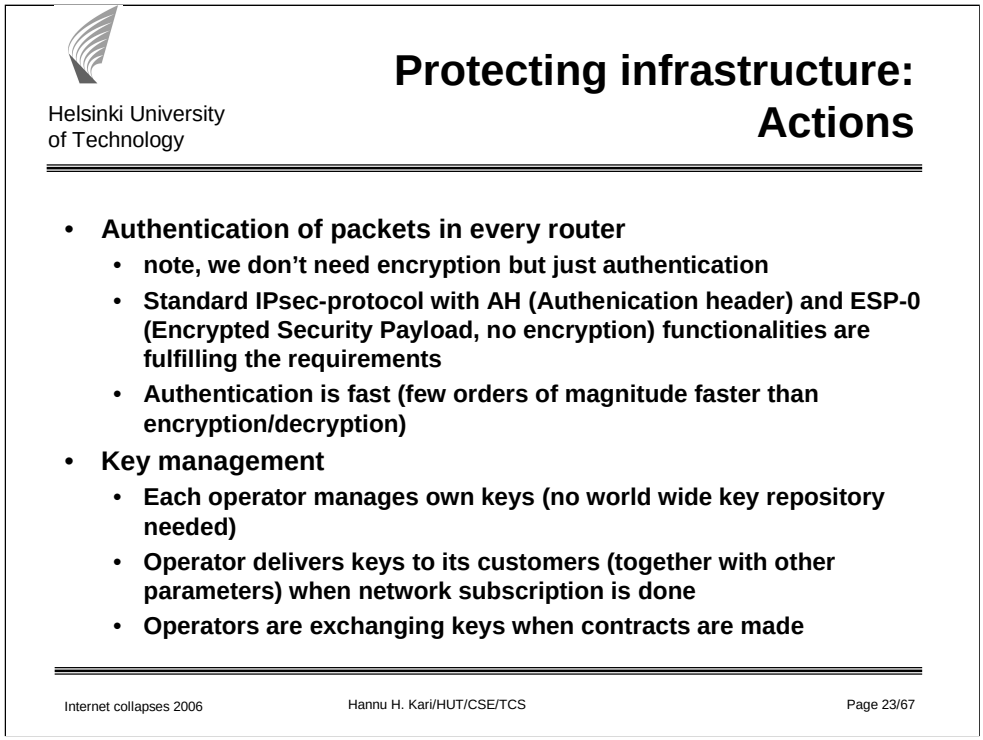


When we want to ensure, that our virtual networks are well protected, we need to have real security solutions in use. This means that we use similar approach like in page 17, where we have both link and path level protection. Then, we can detect unauthorized modification of unprotected virtual network identities.

Both link and path level protections are needed since

-we want to detect promptly the physical attacks (already at the next router)

-we don’t necessarily trust on the network operator(s) (we want to ensure that traffic is coming from our trusted partner SGW)



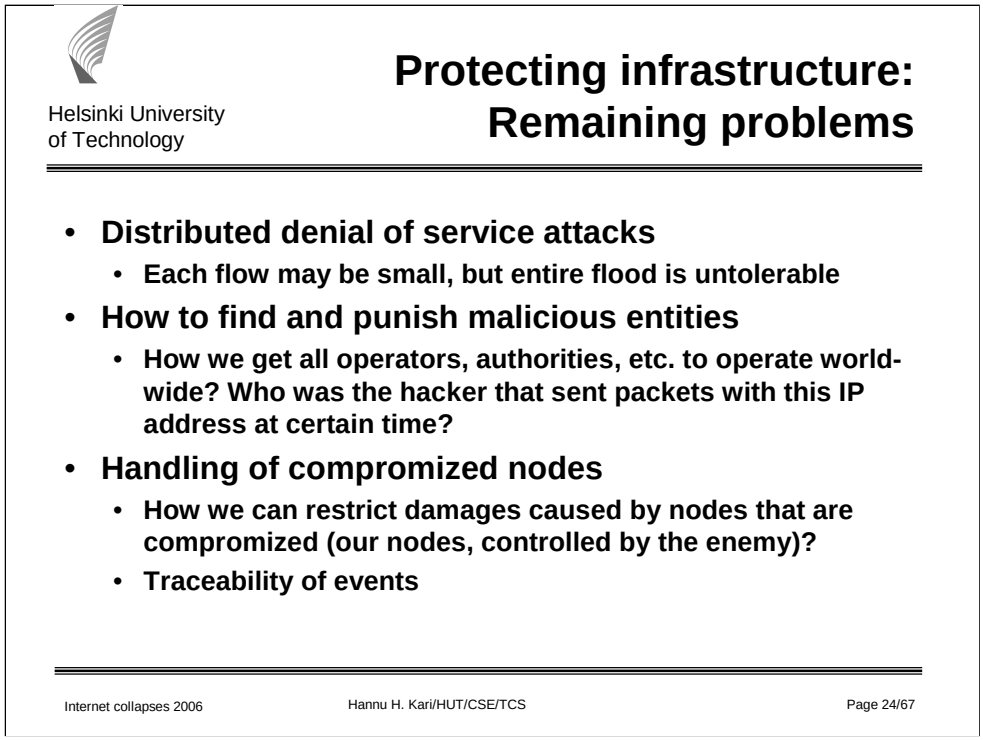
Authentication can be done with standard existing protocols (e.g., using IPsec). Since we only need on this level authentication, not data protection, we can do it efficiently. The main penalty (beside the extra processing in every router) is the extra authentication header in every packet that lowers the channel throughput by about 10% of the channel capacity.

The key management is decentralized.

-Since operator manages its own network and sets up trust relation within its own network, this can be done locally.

-Operators have trust relation only with its own customers. Thus, key management for those customers will be handled during the normal service subscription procedure

-Trust between operators is handled during the normal setup between the operators. Each operator needs to build trust relations with its adjacent operators.



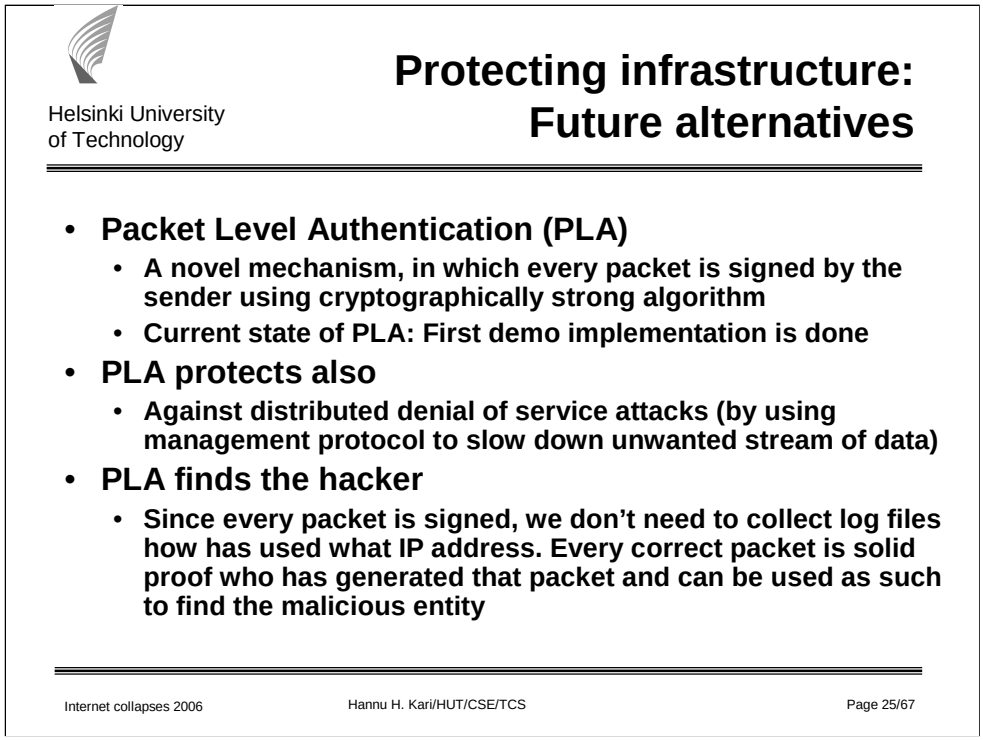
The proposal on the previous pages is not the final solution, but just a quick remedy before we get a real solution. For example, we have still following types of problems:

-Distributed denial of service attack, where enemy uses large number of (innocent) computers

-Since Internet is global, we must have protective actions in use every where, not just in our counrty. Unfortunately, many countries and operators are reluctant of taking any measures to minimize the problems that are caused by users in their network against rest of the world. By threatening to disconnect such operators from Internet seems to be the only solution against such lazy operators.

-Unprotected computers can be easily hijacked by hostile entities. Once such incident happens, we must have means to exclude such computers from the network efficiently.

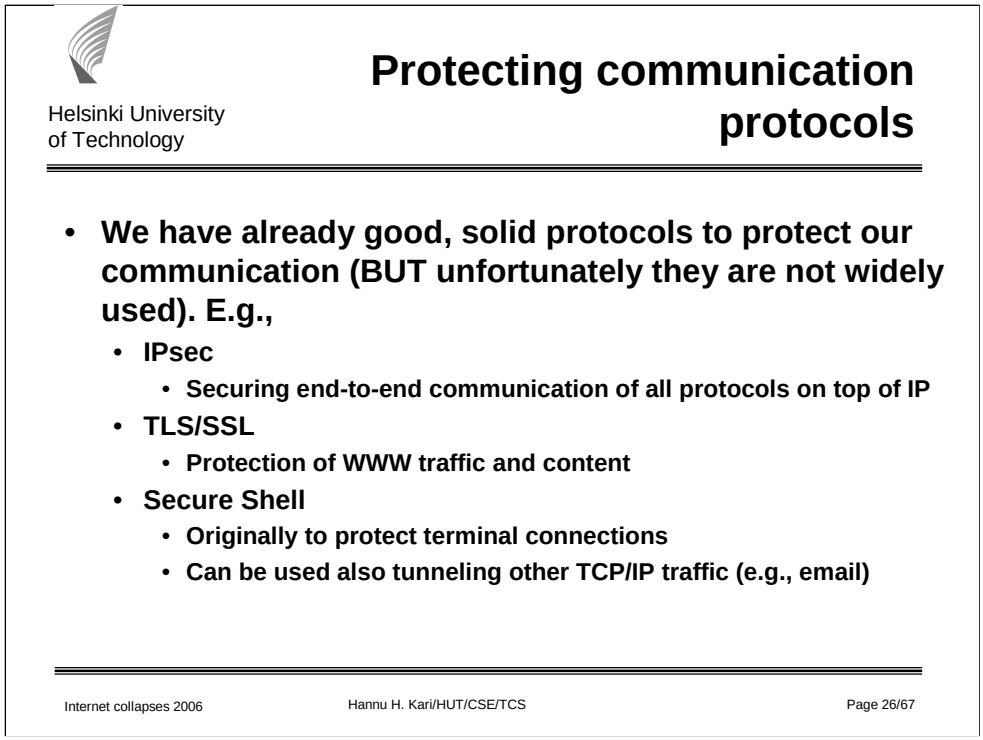
These topics (Distributed DoS and compromized nodes) are solved with Packet Level Authentication (PLA) concept.



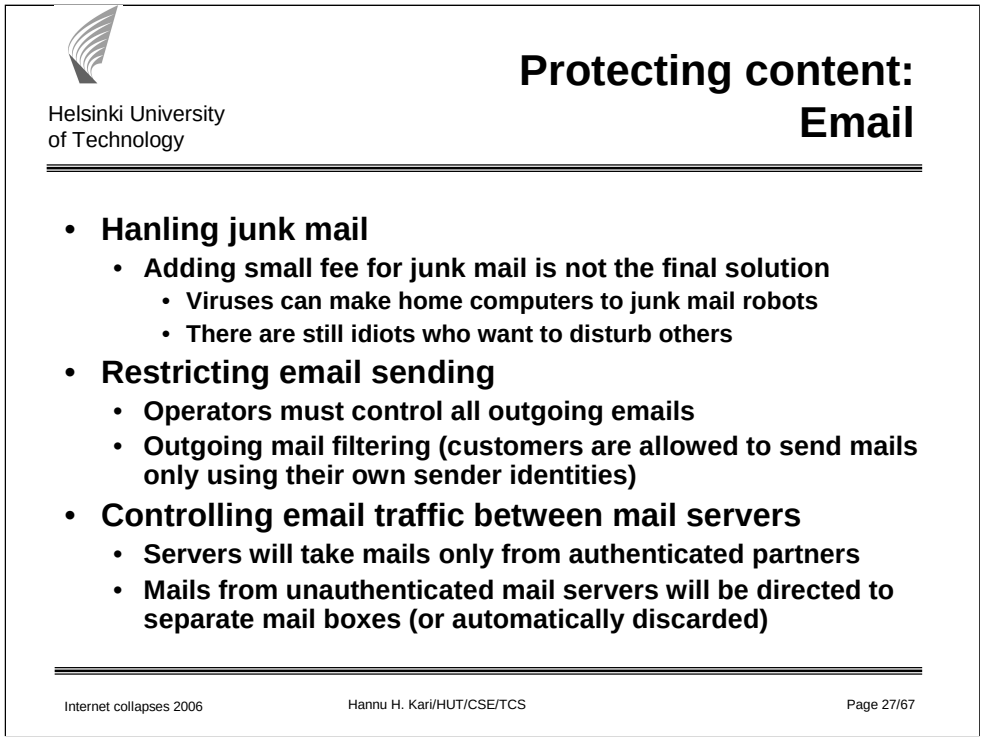
We (at HUT) have developed a novel idea for protecting IP traffic against various attacks. Originally, it was designed for wireless ad hoc networks in military environment, but it will be scaleble also for civilian environments.

The main idea is that sender signs every packet that it sends using cryptographically solid algorithm so that any other node may verify that the packet is OK. Since we use public key –method for signing packets, only that entity that has the private key can sign the packet. Every node that has the public key can do the verification. Since we carry the public key in every packet, any node that receives the packet can also verify the integrity of the packet.

Finding out the hacker in PLA concept is easy, since we take one packet from the hacker and look who owns the public key and then we know who it is...



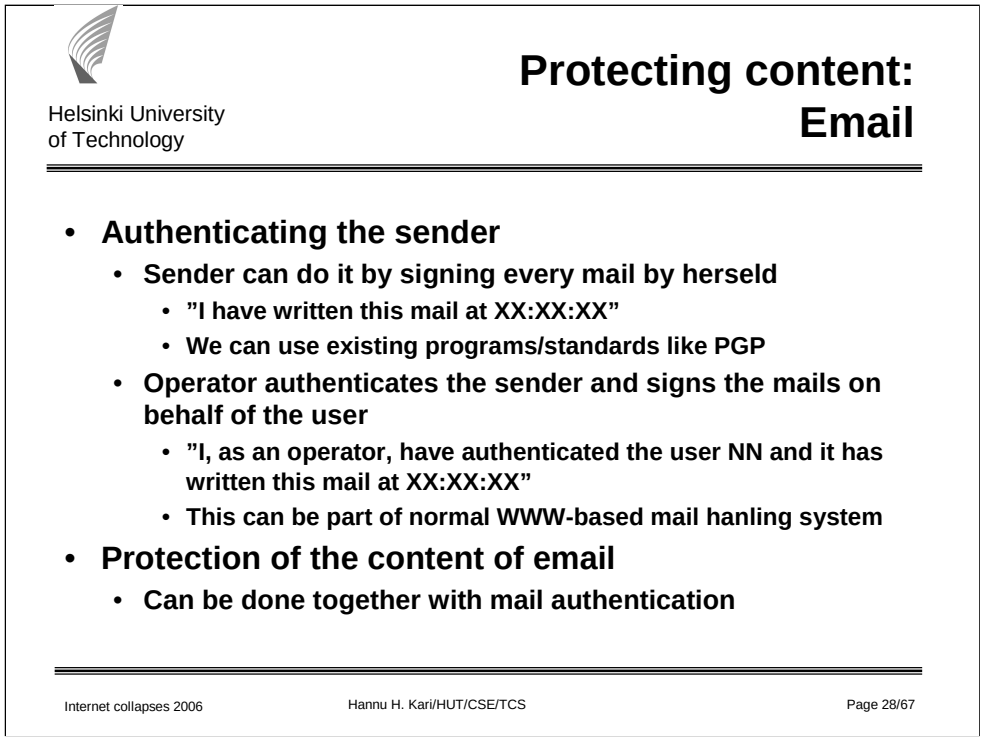
The second level of my model (communcation protocols) is at the best shape. We have solid protocols, but not all of them in use. We can use IPsec to protect all of our communication between computers (or networks). TLS/SSL can be used for protecting web traffic. Secure Shell is an example of applications that allows secured terminal connection over public networks).



Some of my colleagues have proposed as a solution for junk mails to put once cent price tag per mail. Unfortunately, this does not solve the problem, since a home computer (infected with junk mail robot virus) can send one million junk mails per day (i.e., about 10 000 euro per day cost to the home user).

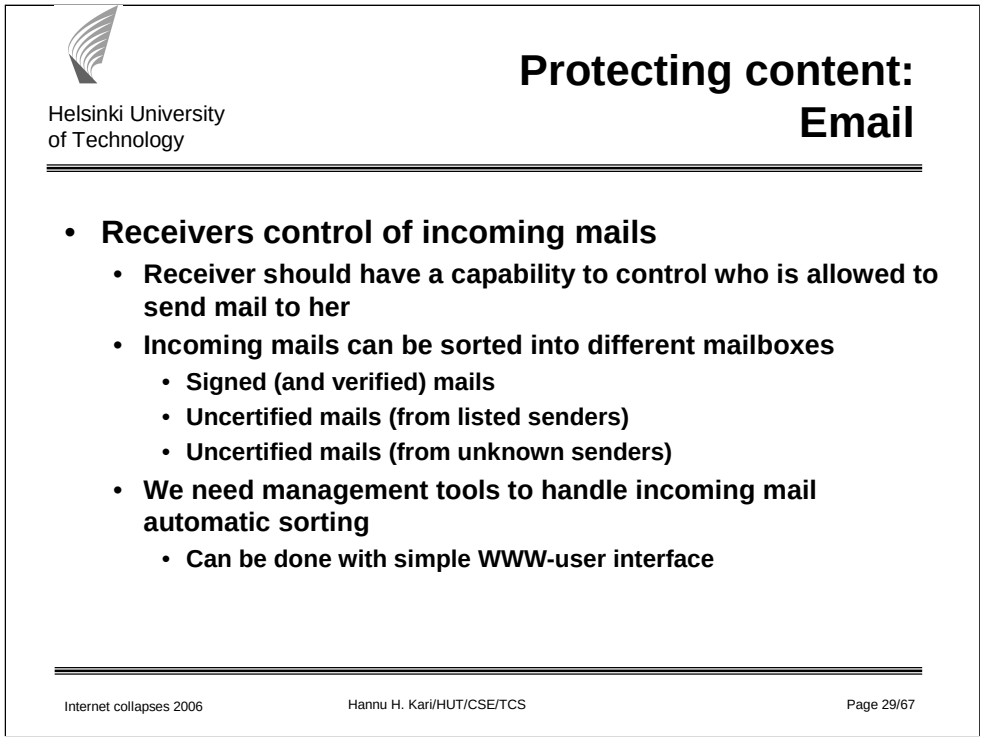
More efficient way of limiting junk mails is to have strict control of outgoing mails. This means that all maill traffic from users must go through operator's filters. Then, operator verifies that the user has legitimate rights to use the sending mail address. Why on earth operator allows users to forge mail sending address at the first place? Especially, since many Internet subscription contract explicitly disallows this!

Also, we need to have protected communication between the mail servers. Then, if operator verifies that sender is rightful user and owns the sender identity and mail servers verifies each other, most of the traffic is already this way at some level protected. All mails that can't be verified or are coming from mail servers that are not trusted are, by default, treated as junk mails.



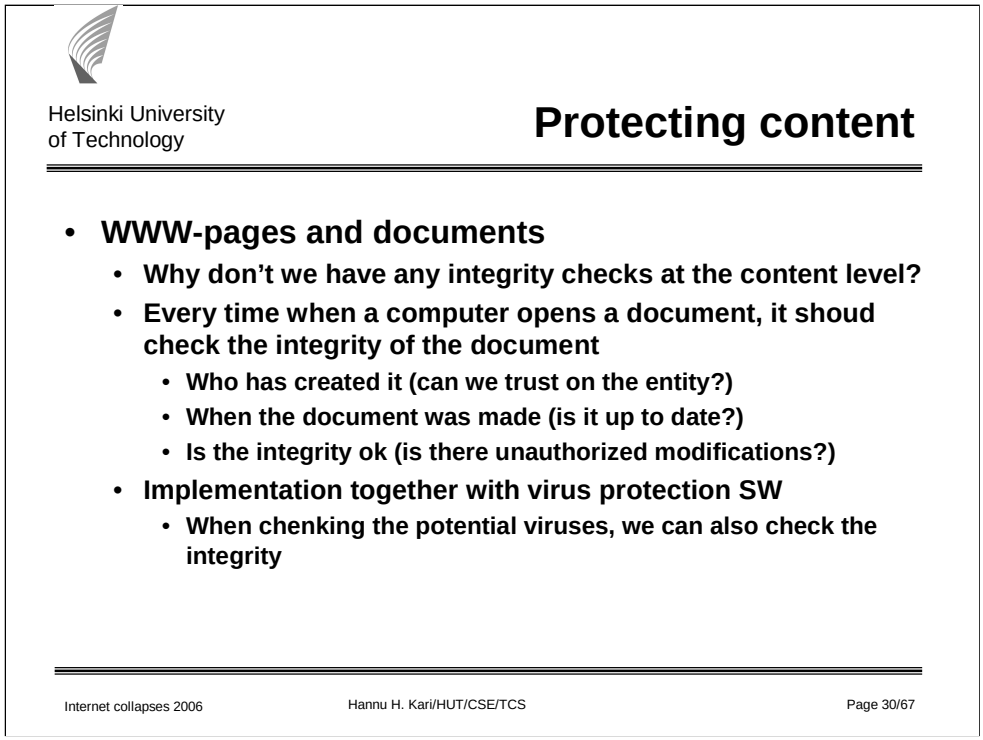
When we really want to have trust on mail system we must have a possibility to authenticate the sender and protect the content of the email. Both of these can be done with existing programs/standards like PGP. With PGP sender can both sign the mail and also encrypt the content.

In case user does not have own computer but uses WWW-based interface to read mails, the operator can do the signing on behalf of the user. This means that receiver will see difference: Instead of ”mail authentic from user X” we get ”mail sender X verified by operator Y”.



The most important thing for the future is to switch the control of communication from sender to receiver. Then, the receiver defines, who may send mails (or more generally, who may contact me) at the first place. Most people are happy to have a list of friends that may send mails, and need a simple tool to manage the list of friends. Public persons, such as a university professor, needs to be capable of receiving mails from previously unknown senders. However, those senders should be trackable (e.g., in case of hate-mails).

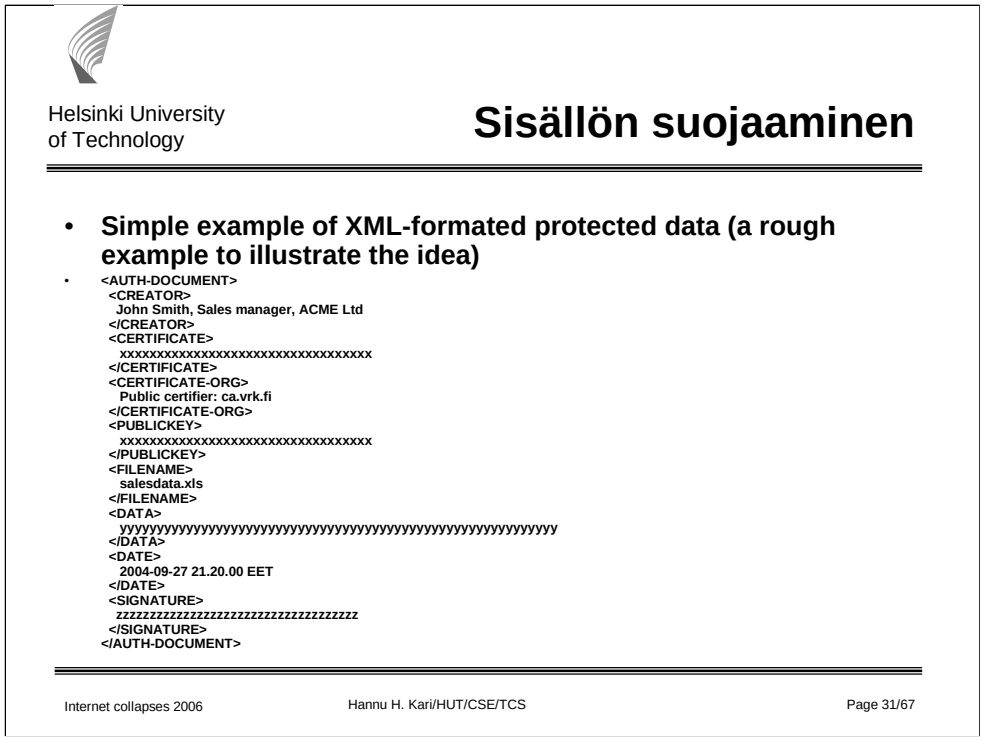
A simple mechanism of mail control system is to sort incoming mails into few different mailboxes based on verification. All mails that are verified can be passed through to the primary mail box. Since not everybody has at the beginning this nice mail authentication system, we may want to sort out certain senders mails from junk (but since those mails are not certified, we should not mix them with certified ones). Finally, all those mails that are coming from unknown origin and are not certified can be treated as junk mails and can be either automatically discarded or stored for short time (”just in case”) before automatically discarding them.



An other important issue on the content level protection is WWW-pages and all other files/document. We should have a mechanism that verifies that the document in use is authentic.

We can't rely on WWW-protocols (such as TLS/SSL), since information may have been modified at the proxies or files are coming via other sources (such as email). Thus, we are proposing alternative (at the next page) that can protect any file or any information.

Checking could be implemented as part of the virus protection program that in any case needs to inspect the file against potential viruses.



This simple example ilustrates the idea how we can protect the files using XML-formats. The structure contains few fields of information that the ”virus checker” reads and verifies before opening the actual data part:

-Creator: who has created this document

-Certificate: Here’s the certificate that ensures the the creator is good guy

-Certificate-Org: What organization has done the certification

-Public key: What is the public key of the document creator

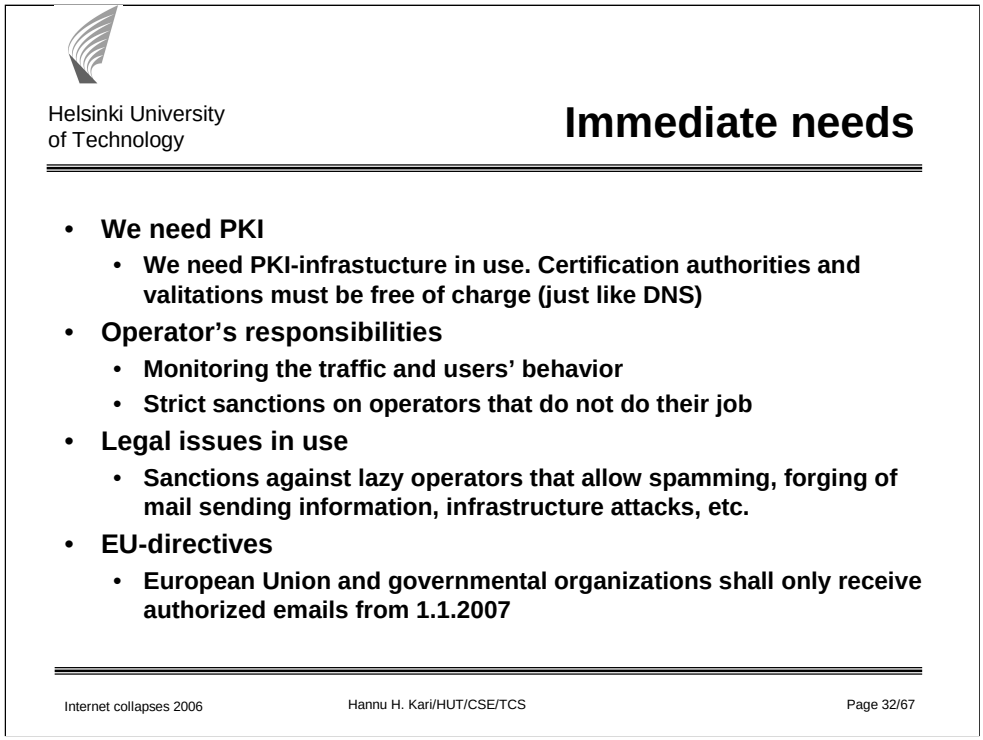
-Filename: What is the name of the file

-Data: Here is the actual content of the file

-Date: When the data has been made/updated

-Signature: And finally the creator’s signature that it has really created this file

With all this information, it is possible to verify the document’s timelyness and integrity (provided that we can trust the certified organization).

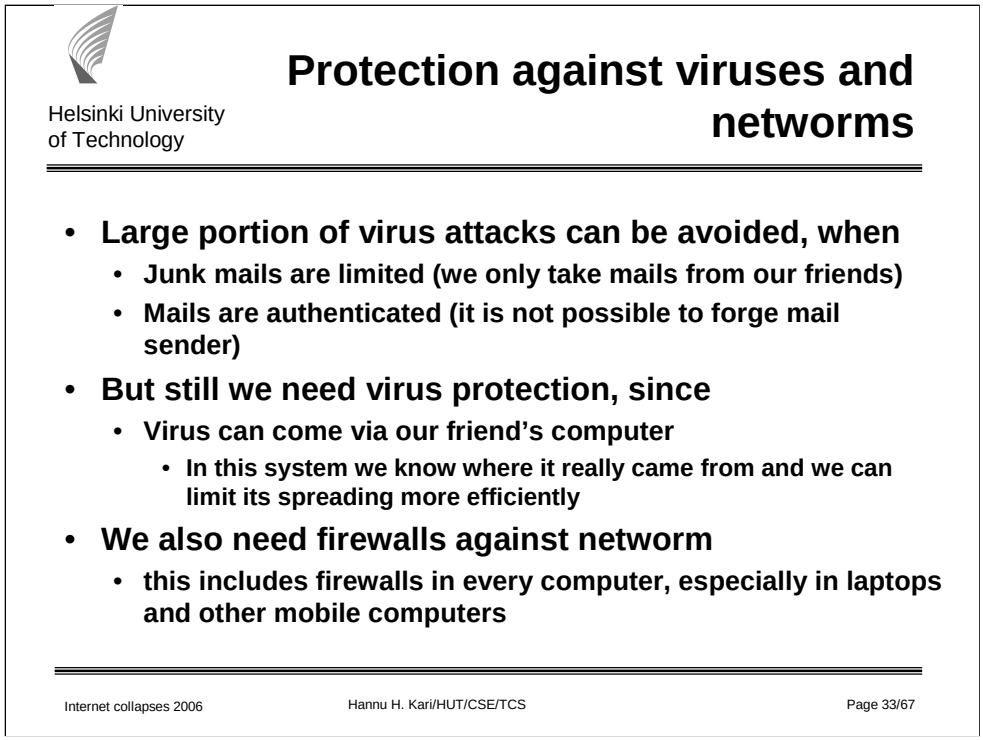


Some immediate actions are needed:

-PKI-infrastructure must be taken into use. It shall be free of charge to use, just like name servers. Otherwise, it will be delayed too much.

-We must take all measures to ensure that operators will do their job. Best way to handle this is to ”hit their pocket”. If their income is in danger (due to financial sanctions, lost revenue, lost customers, etc.), they will react.

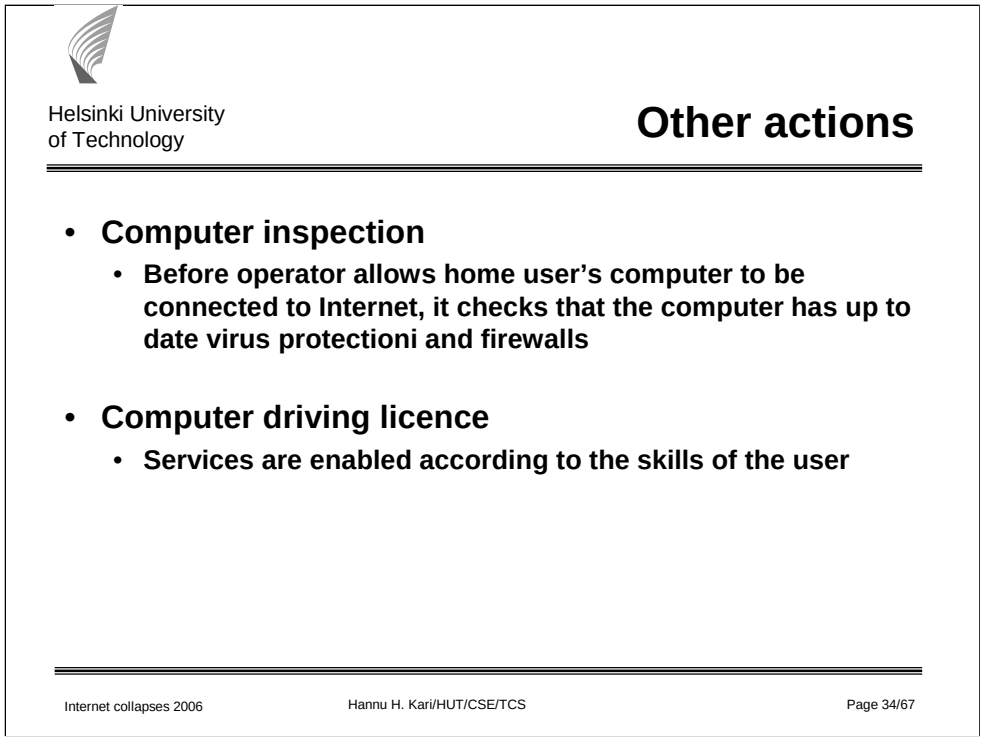
-The easiest way to enforce certified emails is to mandate that on the EU level. This does not mean that we force all companies to use certain mail program or certain operating system (like today some EU reports must be done using Microsoft's Word document). Instead, we specify the mail standard PGP or S-MIME, that allows any program or mail application to be used. Corporations that want to do business with EU or other governmental organizations must sign their mails. This can be done also at the corporate level (if that is the decision of the company).



Even when we get better protection in the network, we still need virus protection and firewalls. All those counter measures presented earlier are just limiting the protential threats but not eliminating them totally.

Fortunately, we can minimize junk mails and also virus mails from hundreds per day to maybe just one per week (that depends on the organization and user behavior).

The most vulnerable computers in the organization are mobile computers (laptops, PDA, mobile phone) that are most prone against various attacks. Especially, because they use wireless links in communication. Nowadays, most laptops have WLAN interfaces that are by default always enabled. Since the laptops are most likely having lower speed connection to the network and significant communication costs, the virus protection programs' latest updates are coming late to those computers. Thus, they are also prone to virus mail attacks.

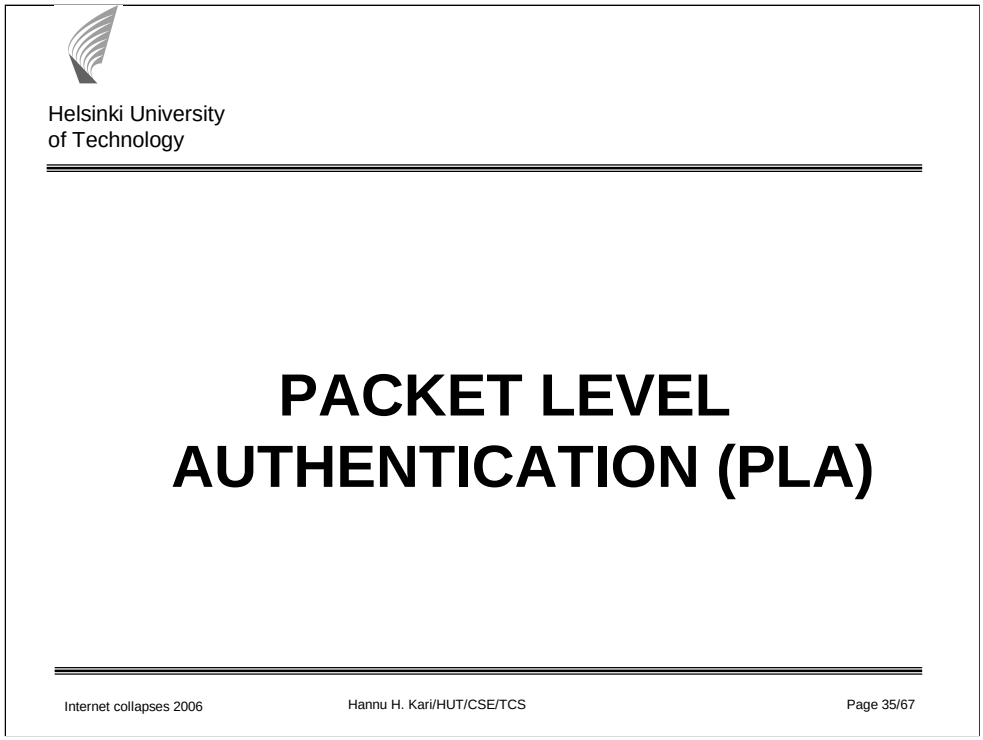


The analogy comes from traffic...

In Finland, every car is inpected annually, to ensure that it fulfills the state reguirements for public safety. Why not we have the same means for the computers? If you try to connect you computer into Internet, your operator will check that your virus protection is up to date, your operating system patches are done, and firewall is in good condition. After that, you'll connected to the network.

In traffic, not everybody is allowed to drive a bus. Similarly, why every Internet user is allowed to do everything (regardsless of her skills)?

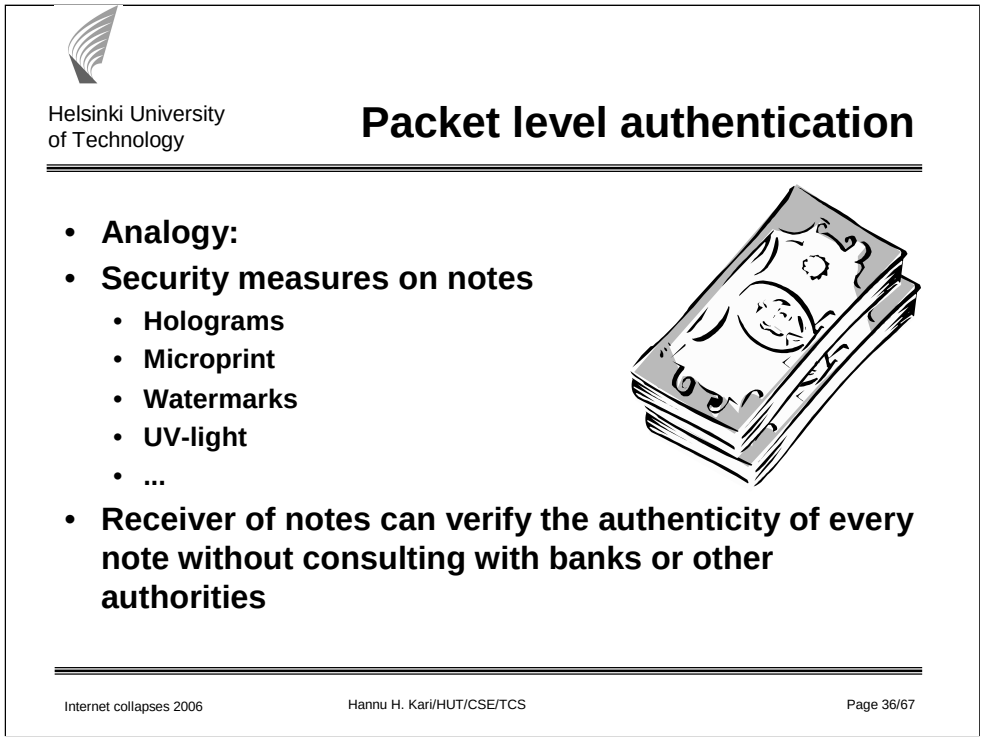
And finally, since many people are not obeying the traffic rules, we need the highway patrols. Similarly, we need Internet police, that monitors our behavor in the net. They shall not be interested the content what we are sending (like the traffic patrol only monitors the traffic regulations, not the content of our car), but how we are following the rules of Internet.



The remaining slides will discuss about one potential solution, that could save Internet from the collapse. At the moment, PLA is at the very early phase, and still requires much work. However, the first proof-of-concept implementation shows that this is doable.

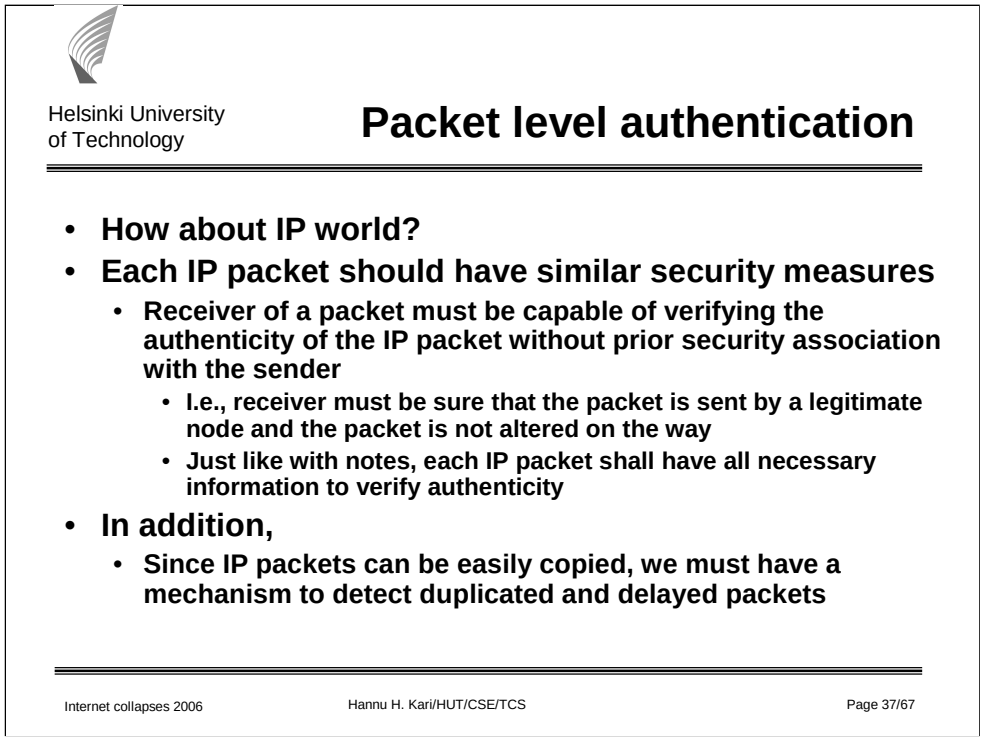
Short introduction on PLA is availabel at

http://www.tcs.hut.fi/Software/PLA/



The original idea came from the analogy of notes. As there are large number of forged notes (especially US dollars) in circulation, there is urgent need for detecting real notes from forged ones. Traditionally, there are several security measures that makes forging of notes difficult (such as holograms, microprints, watermarks, etc.).

All these security measures allows the receiver (e.g., a shop assistant) of the note to verify the authenticity of the note without need for consulting the ”final destination” of the note, the bank). This is important, since the shop gives some service to the customer immediately.



Similarly, a router, that gives service for the IP packet (or the sender of the IP packet), should be capable of checking whether the packet is authentic or forged. In the latter case, the packet should be discarded immediately and an alarms should be risen. But in the former case, the service (i.e., routing packet further) should happen with minimal delays.

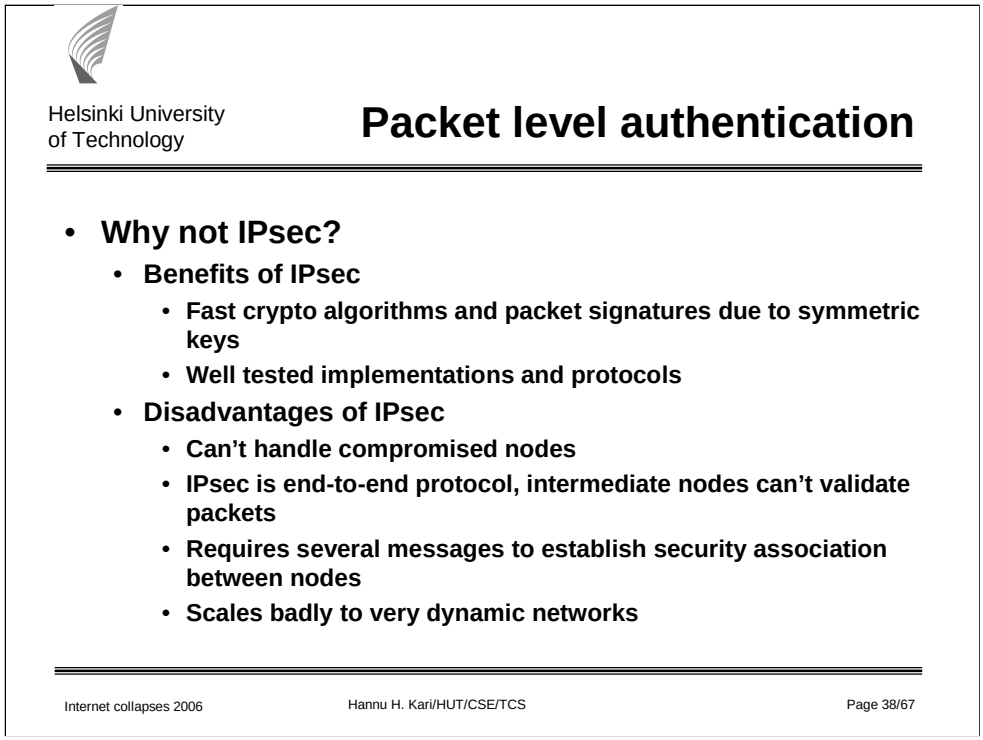
Since the packets are very similar like notes, independent of each other, we should have a mechanism that allows us to treat every packet individually with prior negotiation with the sender.

Additional problems are coming because of the difference of physical notes and electronic bits:

-It is impossible to make identical copies of the note, but bits can be easily copied as many times as wished

-Once you give away a note, you don't have it, but bits you can keep copies as long as you like

Thus in addition, we need detection of duplicate packets and packets that were sent long time ago.



Next question that comes is, obviously: why don't we use existing security protocols, such as IPsec? IPsec is good for protecting data confidentiality and integrity end-to-end. It is based on well tested protocols and commercial implementations are available.

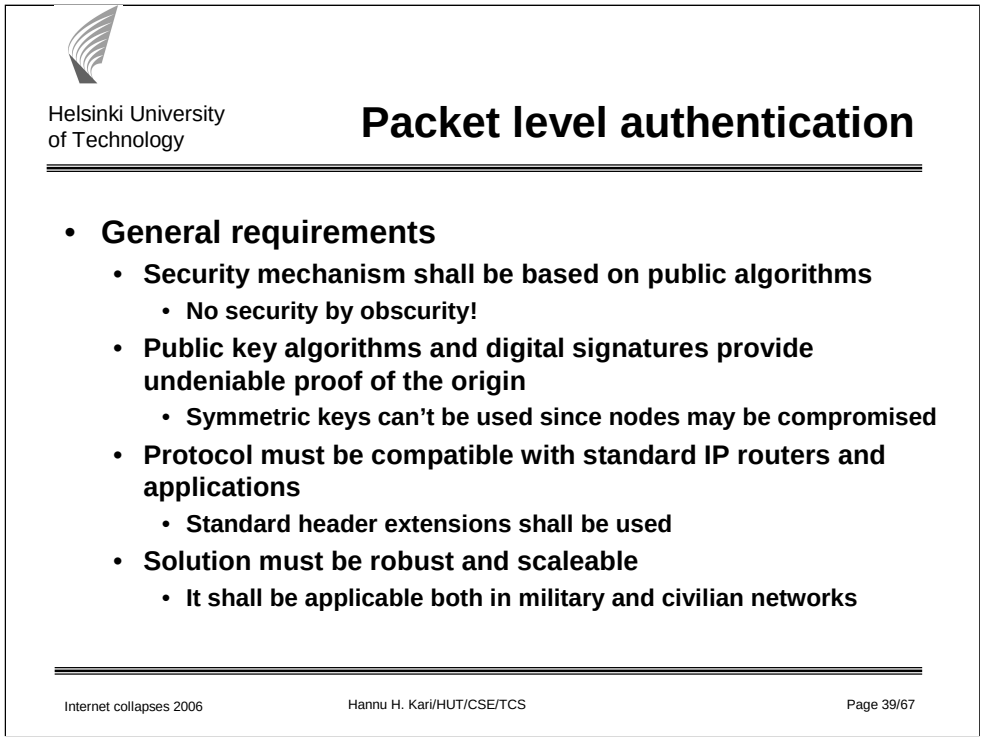
But IPsec has problems:

- IPsec requires several messages to establish a security association between the sender and receiver. This is not always possible in a very dynamic environment.

-Intermediate nodes are not capable of validating the packets. In order to validate IPsec's MAC, every node needs to get the authentication key. Since the same key is used in signing the packets, this means that any node that is capable of validating packets, can also modify and re-sign the packet.

-IPsec has not good methods to handle compromized nodes.

Thus, we have created a new protocol: Packet Level Authentication (PLA)

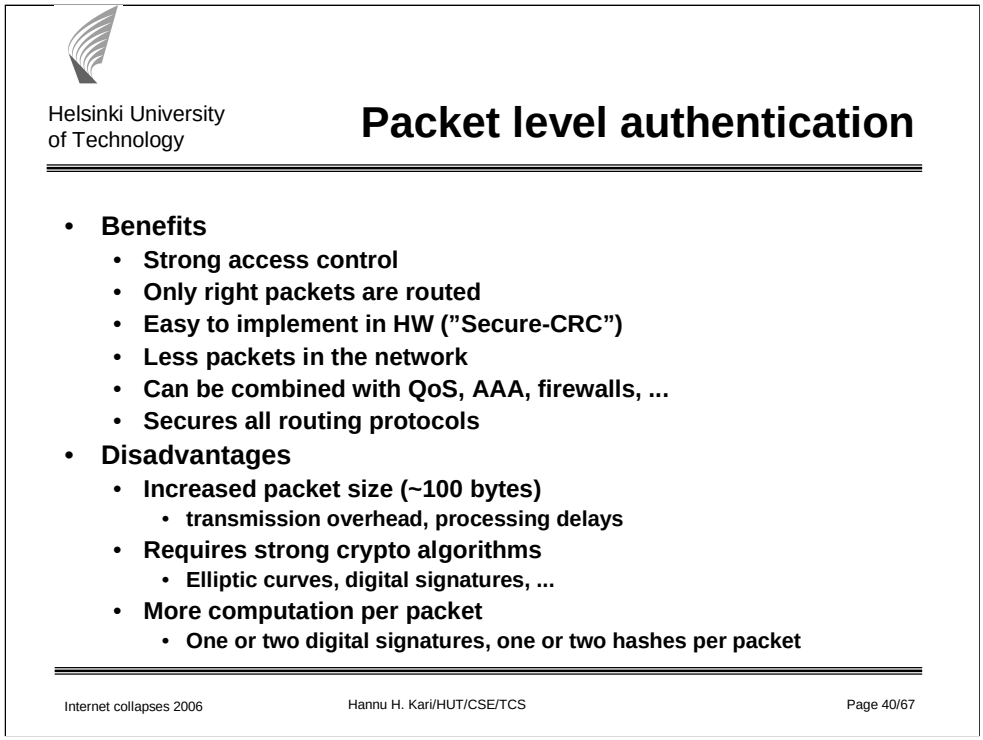


PLA is based on open protocols. Unlike GSM (or other telecom systems), everything is public in PLA and its strength is not secrecy but computational complexity.

Since PLA is using public key algorithm, only the entity that known the private key can sign the packets. All others that has the public key can do the verification. Like stated earlier, we can't use symmetric keys since there would at least two places where we have the same key, and if either of them is compromised, ... Also, in case of symmetric key, we can't be sure, which one of the key holders has signed the packet.

Mandatory requirement for PLA is that it interoperates with other protocols and can be used also in a network that has non-PLA routers. This can be done easily by introducing a new IP-header externsion. Similarly like Mobile IP and IPsec, are adding extra header into an IP packet, also PLA adds its own header. Then, standard routers can ignore PLA header and forward PLA-secured packets as any IP packet.

Design goal is to make PLA scaleable from military networks (where wireless bandwidth is very limited and security threats imminent) to public Internet (where large thorughput is needed in core networks and number of players is large).



The main benefits of PLA are

-Since every packet is signed, we can use that also as strong access control method (no need to do any other authentication and access control measures)

-Since we can discard all malicous packets, we minimize the traffic in the entire network. This is important especially when we are under attack

-PLA can be combined with QoS, access control, firewalls, etc. as, for example, nobody can send packets on behalf of others since forged packets are easily detected. As PLA also has replay protection, nobody can cause problems to the legitimate node by replaying its packet

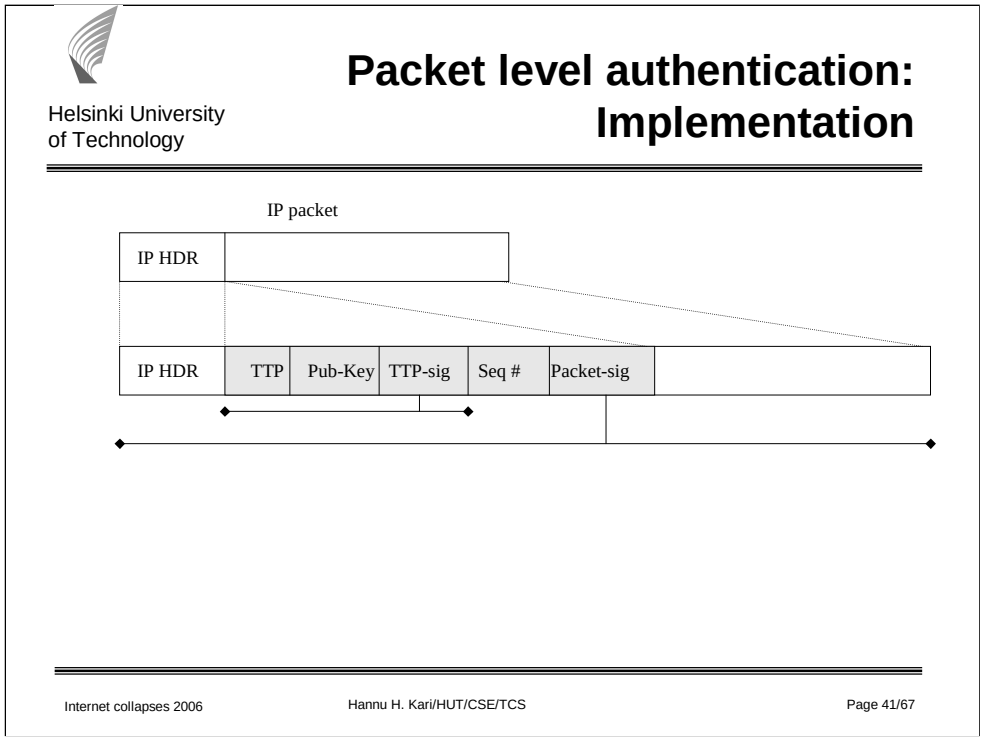
-In ad hoc networks (where ad hoc routing protocols are insecured), we can use standard ad hoc routing protocols without comprpmising our security

-HW implementation is preferable and can help us much on the scalebility issues

And it has also dome disadvantages:

-Obviously, public key algorithms are requiring lots of computation and traditional algorithms, like RSA, have long keys. Thus, we should use algorithms like elliptic curves, that have much shorter keys.

-In practice we need two signatures per packet, that increases even more the packet length and computation.

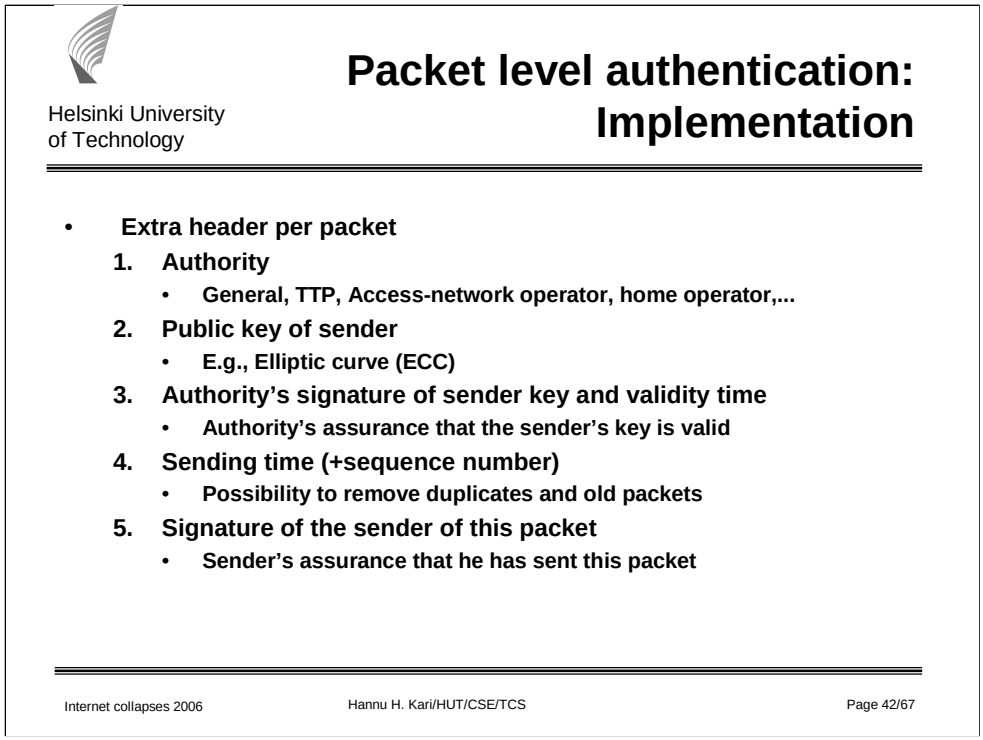


This is a simple example, how PLA works. We add into a standard IP packet a new header, that contains the PLA information.

PLA contains two signatures:

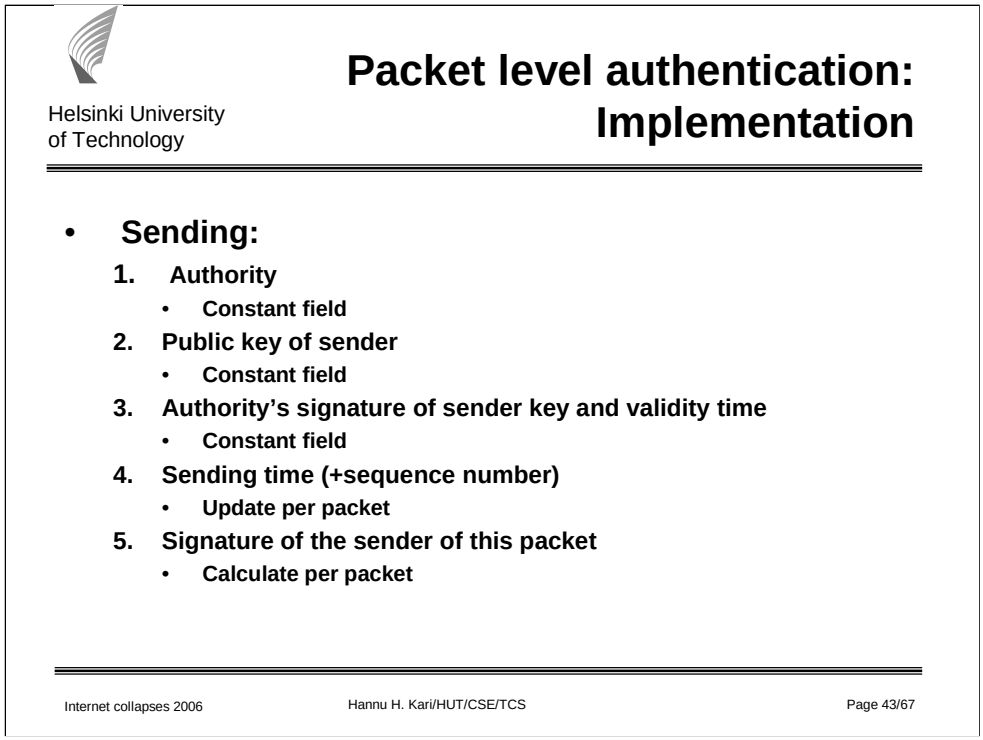
1) The first signature is used to authenticate the sending node's public key. TTP-sig protects TTP information and sending node's public key. This is signed by the authority that trust the sending node and guarantees that the sending is treaceable, if needed.
2) The second signature protects the integrity of the packet. The sending node guarantees with its signature, that it has created the packet and verifier can be sure that packet is not altered on the path.

Certain fields (such as TTL) of the IP packet must be omitted from the signature (just like in IPsec) since they are manipulated on the intermediate routers.



More detailed description of the fields is here:

1. Authority identifies that trusted party that has authorized the sending node. This identity can be hash (or other identity) of TTP, that allows the verifying node to identify the TTP and get TTP's credentials.
2. Public key of the sender is also included into every packet, since the sender has no knowledge whether this packet is travelling via the same path as the previous packets. Hence, every packet may be routed via different routes.
3. Authority's signature ensures that the public key is "a good guy" and can also specify validity period of the key. For example, in Internet, we may have short living validity periods.
4. Sending time and sequnce number are used to detect replay attacks and duplication of packets. In case upper levels are retransmitting data (e.g., TCP retransmit), PLA considers those packets are new packets and increments the sequence number and uses the current time.
5. The last field in the PLA header is the sending node's signature over the entire content of the IP packet, protecting also the IP headers.



When the sending node is sending an IP packet, the following things happens:

-Node adds to the header its authority information (that is contant)

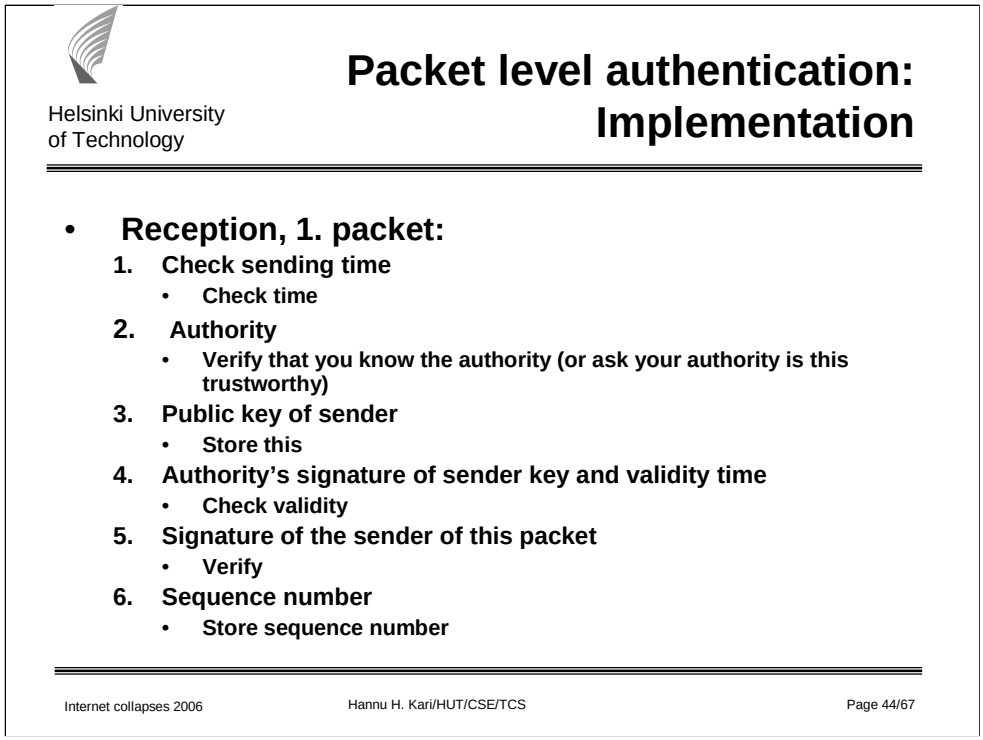
-Also its own public key is added (also constant)

-And authority's signature (also constant)

-Then the sending node takes the current time and incremented sequence number and put them to the header

-And finally calculates the signature over the entire IP packet and puts the result to the header

These operations are basically done just before the packet is sent, so the sending time is pretty accurate.



When any node in the network receives the packet (first packet from this sender), the following things happen (the order of checking can be optimized so that clearly forged packets can be excluded quickly and without much computation):

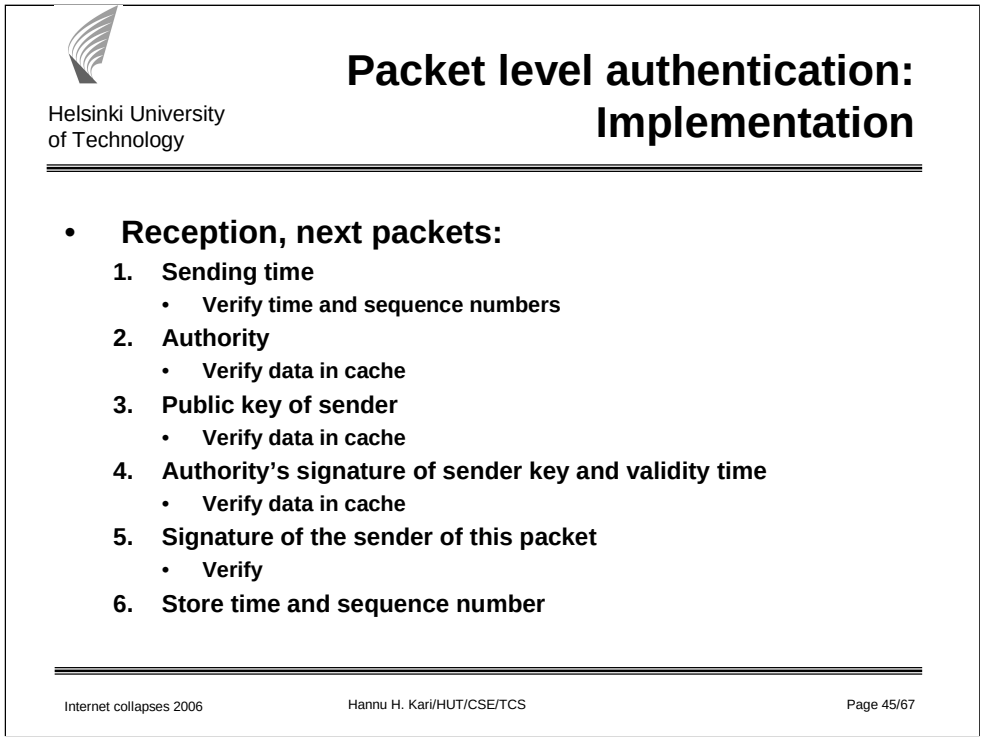
-First the node checks the time when the packet is sent (in military networks we can assume clocks to be synchronized, but in Internet timing must be done per sender). If time is too old, packet is discarded

-Then verifier checks the authority. If it knows the authority already (it has authority's public key), it can then validate the signature. Otherwise, verifier needs to fetch authority's public key from its trusted repository.

-Senders public key is stored, if the authority's signature is correct and validity time is OK

-Then, the verifier knows that it may trust the public key in the packet and it can use that key to verify the packets integrity.

-Finally, the sequence number is stored in order to detect replay attacks and delayed packets.



For the next packets from the same sender, the verifier does not need to re-calculate authority's signature but verify that its the same as in cache.

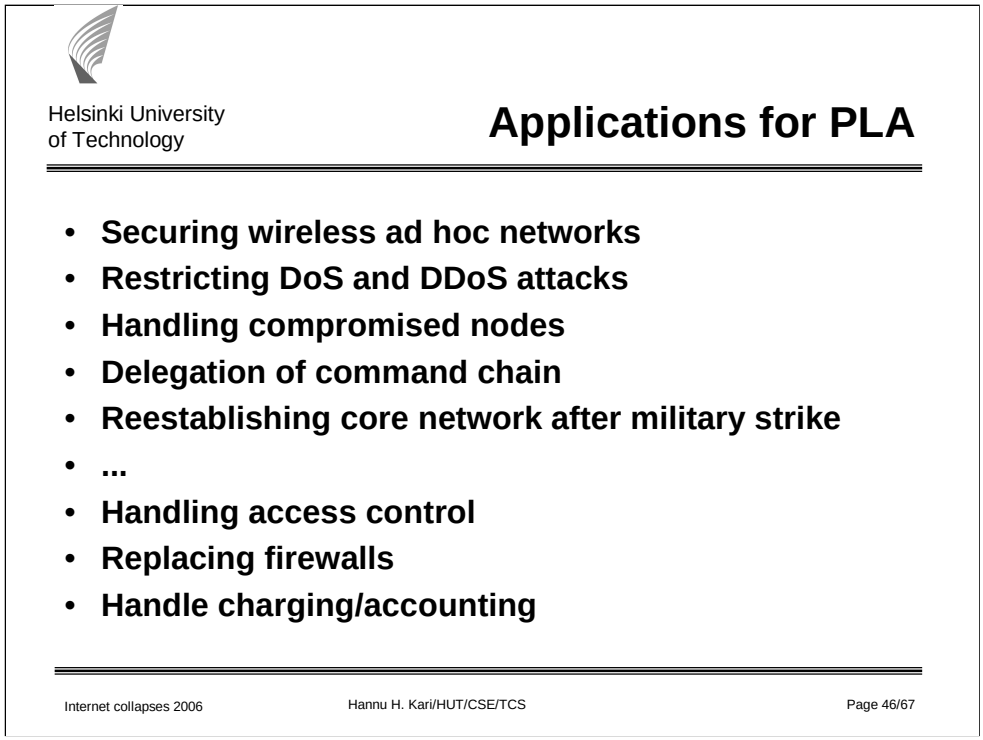
-Again, we start first by checking the timeliness of the packet.Also the sequence number is compared with the most recent packets of the same sender.

-Then Authority, public key and authority's signature are verified with the data in cache (in practice we hash those fields and if we find correct entry, we are satisfied).

-Then, the packet level signature is verified

-And finally we store the time and sequence number for further packets.

In case the verifier is incapable of storing information of all sending nodes in its buffers, it will not lead into serious problems, since the verifier only need to perform revalidation of authority's signature. If the nodes are not having syncronous clocks, then verifier may pass some replay packets.



We can use PLA for various environments or means:

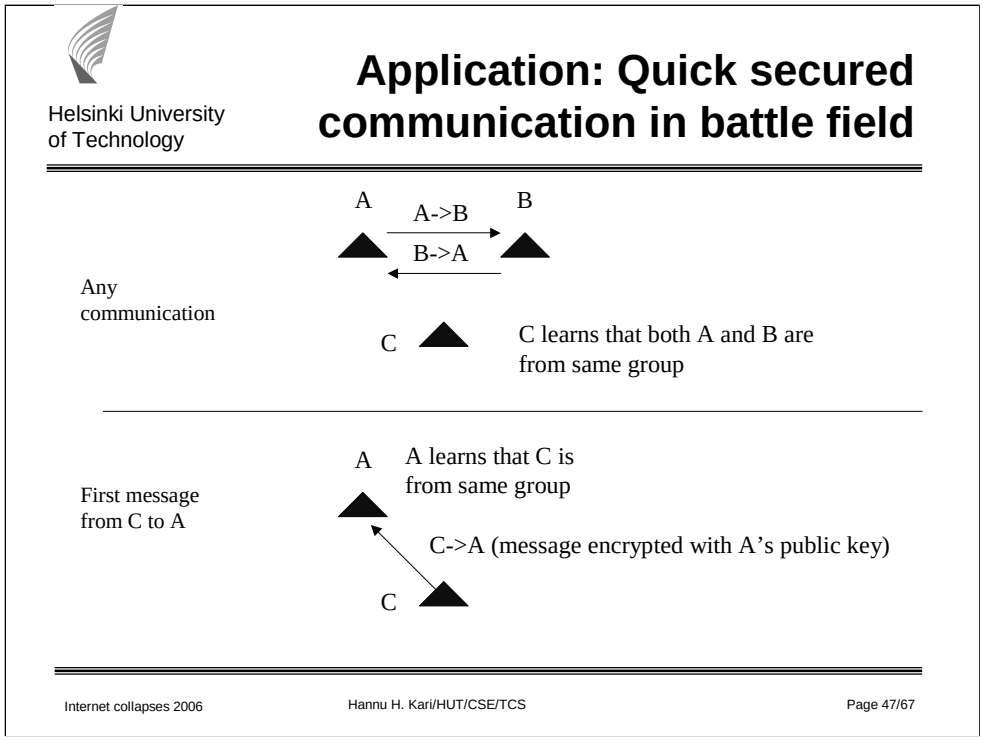
-Originally, it was designed for wireless ad hoc networks for military. Therefore also handling of compromised nodes is built in as well as some other military applications. These applications are described in details later on.

-It also restricts nicely DoS and DDos attacks. For this, we need simple reporting mechanism in which the server under attack sends to upstream one message per attacker: ”Please, I don’t like such traffic, please stop”. Every router on the path can verify that this report is authentic and can limit the traffic, but preferably this is done the access router next to the attacker.

-Operations of firewalls can be simplified for mobile computers since the computer can be identifies as with its public key and the firewall only needs a list of trusted public keys.

-Access control in public Internet access can be handled also with PLA. If the mobile node is validated by the access operator, then the first packet that the node sends contains already enough information to allow the access operator to pass that packet further (”yes, I have authorized this node, and packet is valid”)

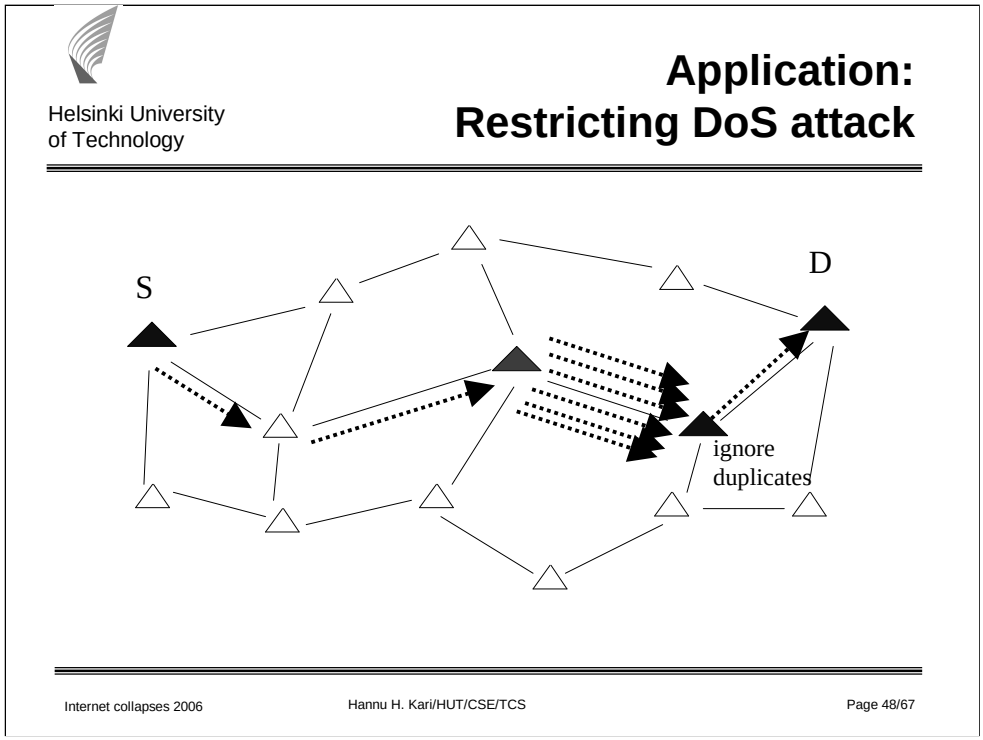
-Afterwards, we found out that PLA could be used also for charging. For example, if the sequence number is incremented with the number of bytes in IP packet (not by one), the sequence number gives undeniable proof how many bytes the node has sent.



Here are few examples, what we can do with PLA.

The first example is from the battlefield:

When a node C hears communication between A and B, it can verify that both A and B are from the same group (or the same side). Then, C can send a packet to A containing all necessary information to establish secured communication. Since A can verify that the packet comes from a node that belongs to the same group, it can further process data. At the payload, there can be information that is already ecrypted with A's public key. So, the first communication packet between A and C is already secured.



Second example illustrates how easily PLA handles DoS and replay attacks.

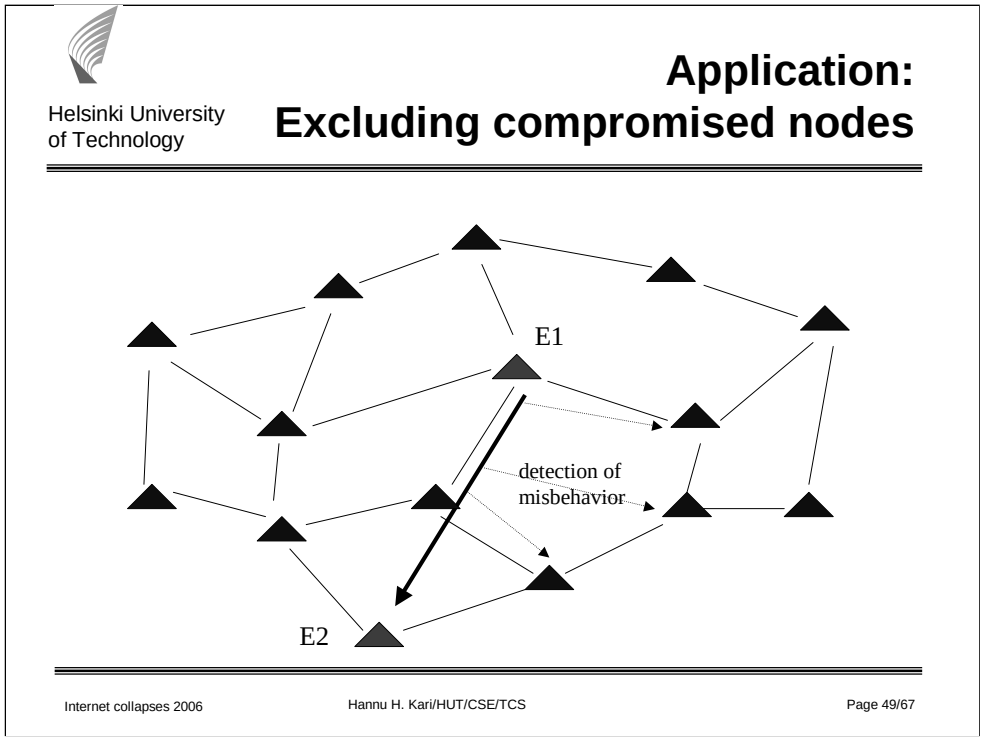
The malicious node (the red on) sits on the path from S to D. Instead of forwarding the normal traffic, it make zillions of copies of the same original packet. The next router will detect immediately the duplicates, since the attacker has two options:

-make identical copies, and then the next node detects duplicate sequence number

-increment the sequence number, and then the next node notices that packet signature is incorrect.

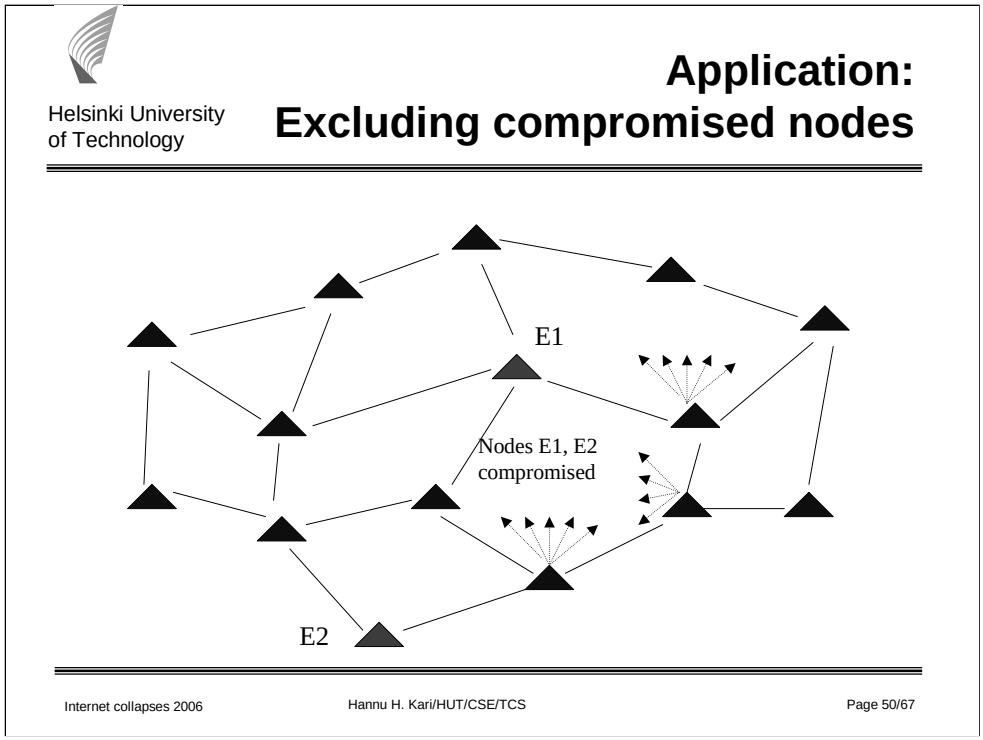
Third alternative would be to send the packets via other paths to the destination. This allows the attacker to make as many duplicates as there are totally separated routes (since any node on the path that is same, will detect the duplicates and ignore them).

Hence, networks, where PLA is in use, malicious nodes can’t use our nodes in flooding the network.



Next example illustrates exclusion of compromised nodes.

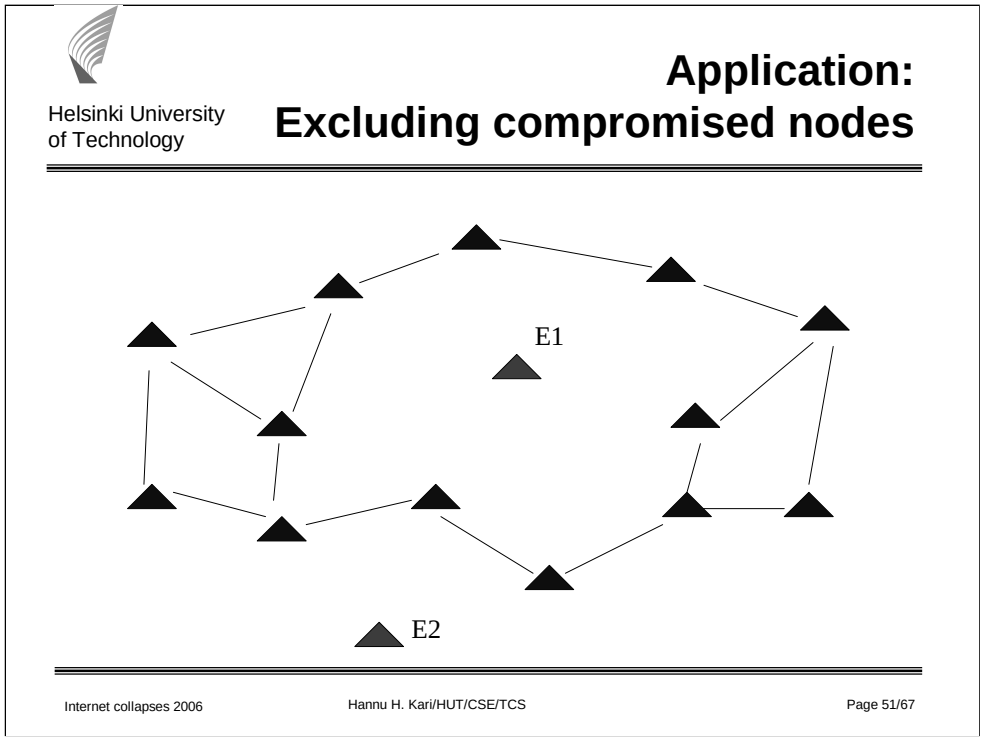
In this case, our nodes detect that nodes E1 and E2 are behaving maliciously (this can be done using a concept of ”Incomplete Trust”). The way how the malicious behavior is detected is outside of the scope of PLA.



Once the malicious nodes are detected, the information will be distributed to the rest of the network that nodes E1 and E2 and not any more trustworthy. This means two things:

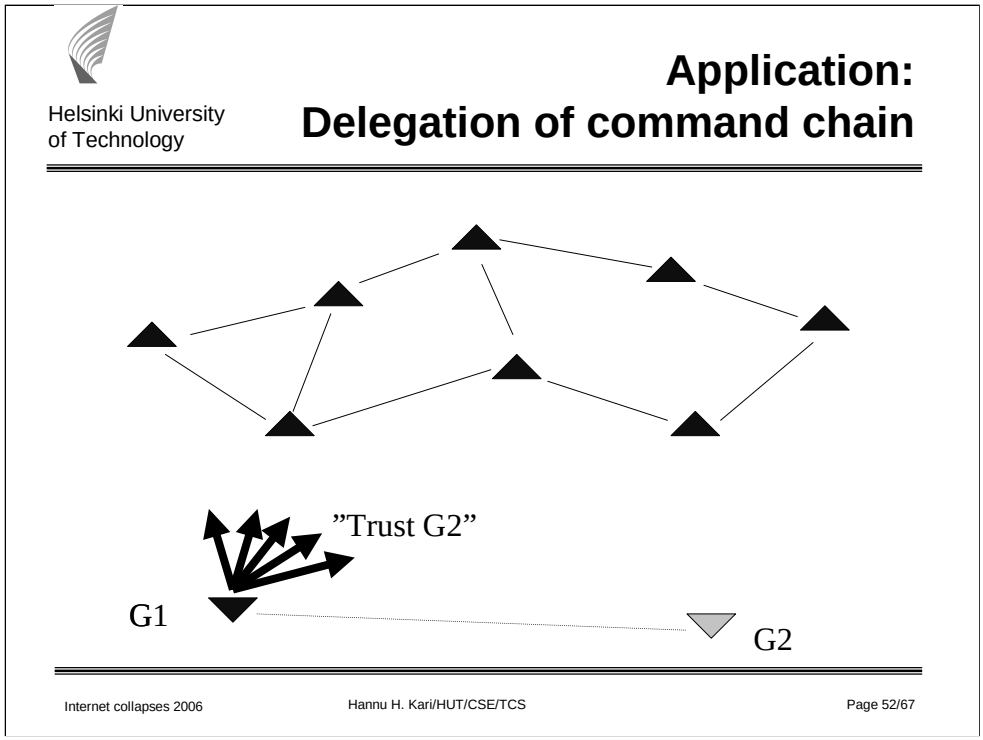
-their certificates will be revoked (if we use revocation lists)

-the certifying authority is informed and those nodes will not be given new certificates



Thus, after a short while, the nodes E1 and E2 are excluded from the network since nobody any more are trusting them.

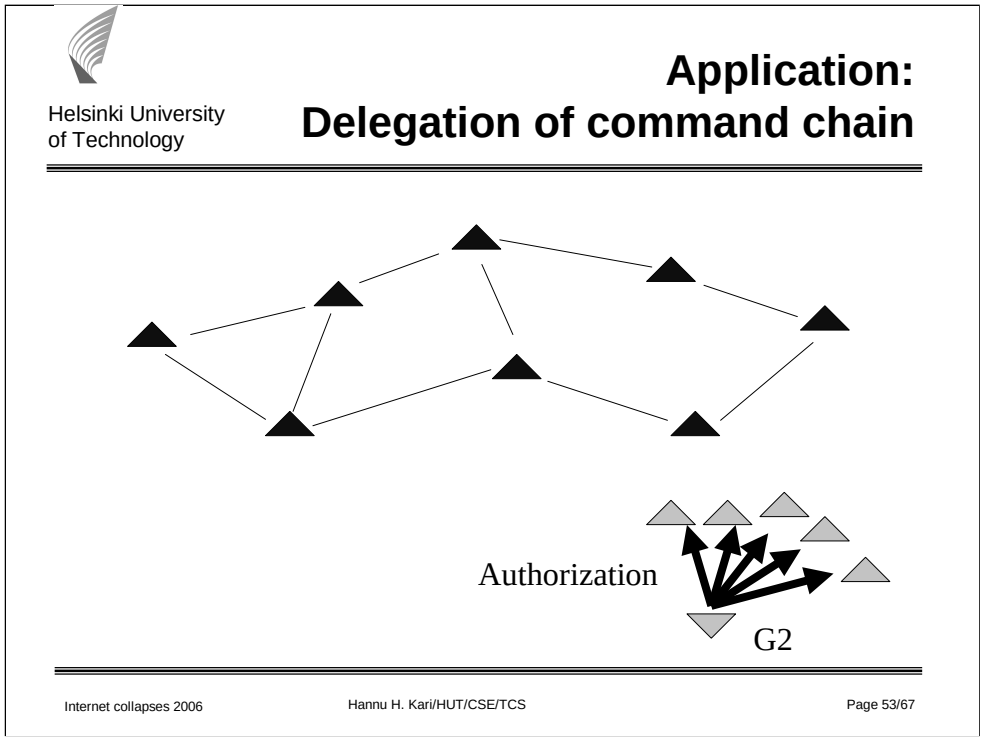
They can still jam the radio network around their cell radius, but they can't use our nodes in forwarding the jamming further in our network.



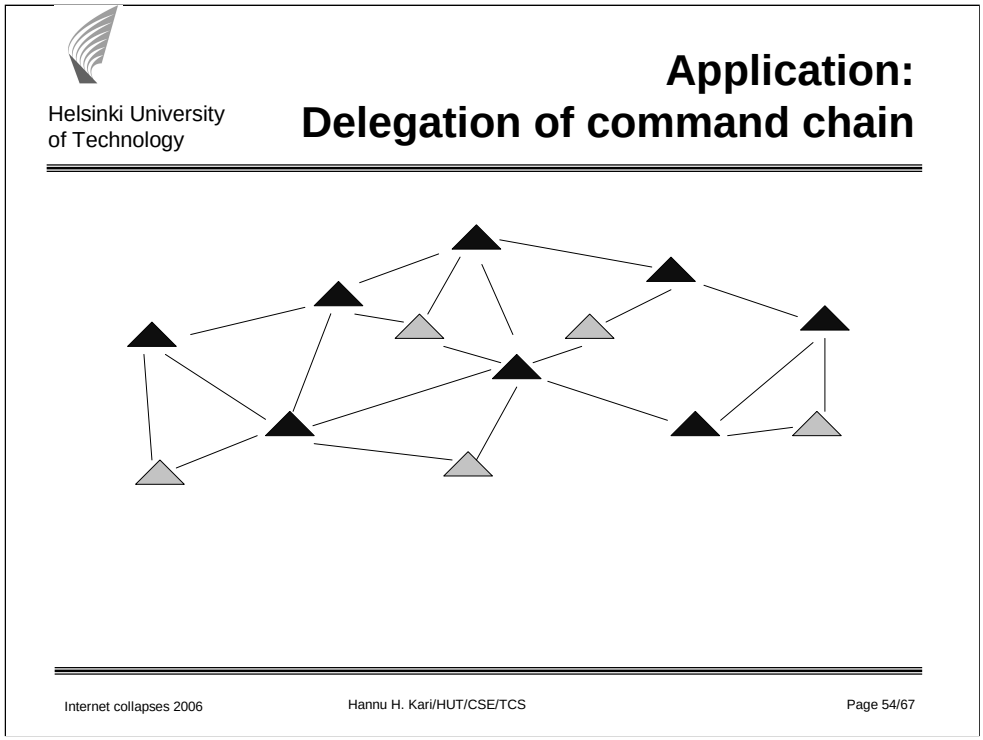
Yet an other example.

This illustrates how easily we can combine and split trust between organizations. In this example, we have for the Blue group (authorized by Blue General G1) controlling peace keeping units of the area.

A new group will then participate the operation, and G1 informs with a signed message, that there is a new trusted third party, Green General G2.

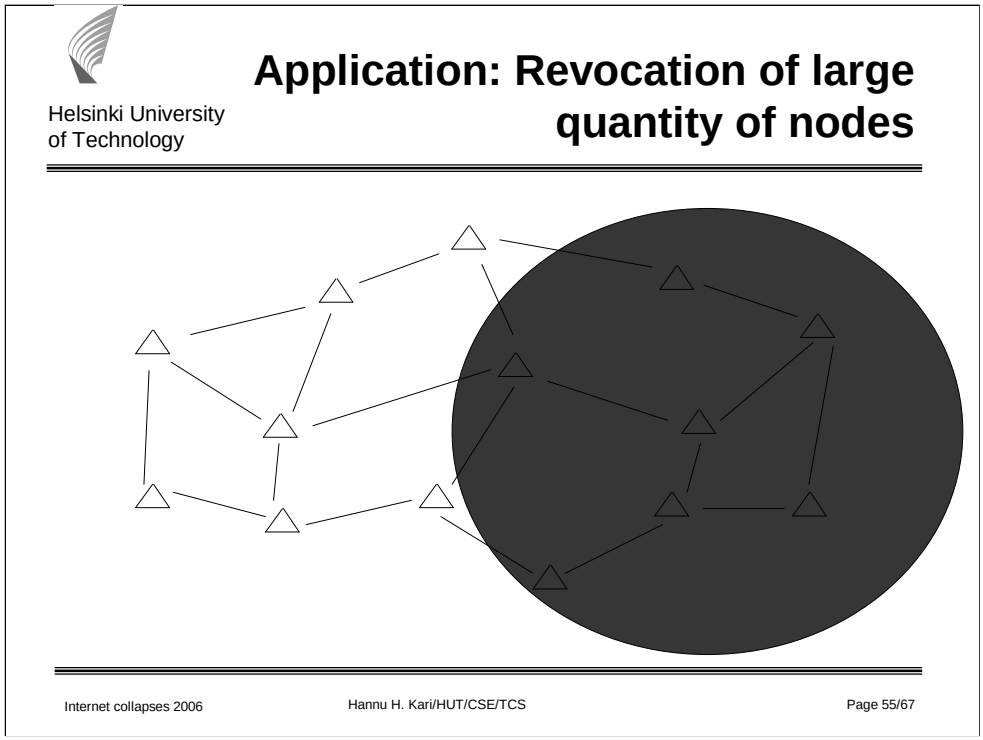


Then, the Green General G2 authorizes its nodes (plus also tells that they must also trust Blue General G1).



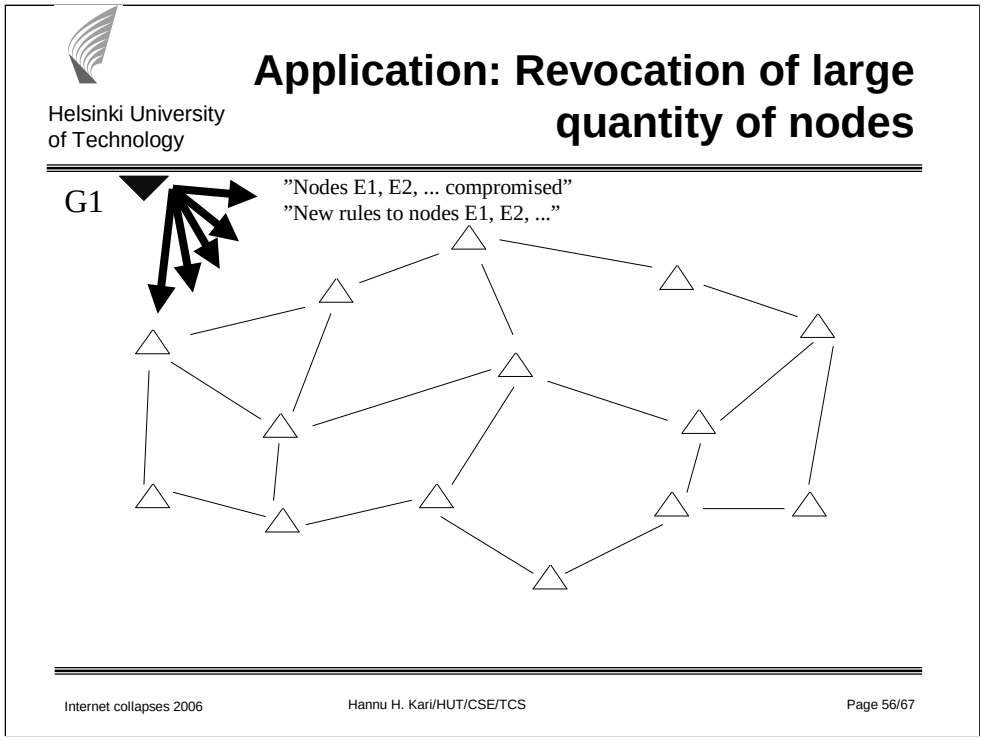
After this, the blue and green nodes can merge on the field. Since they have mutual trust on G1/G2, they are allowed to communicated with each other.

Once the peace keeping operation is over, Blue General G1 [Green General G2] will send its troops a message: ”Stop trusting G2[G1]” Then, automatically, all communication with the other party stops.



Still more examples:

In this case, the enemy has occupied our area (or otherwise take into posession) large number of our nodes.

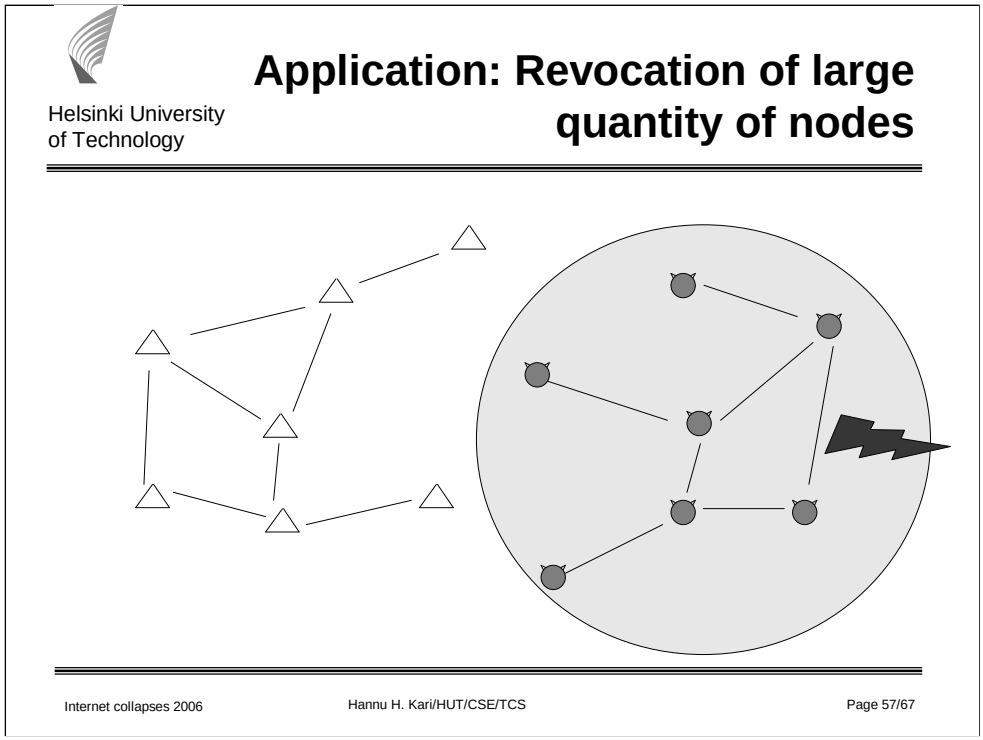


As soon as this has been verified, we can use wide coverage broadcast channel to send two types of information signed by our Blue General G1:

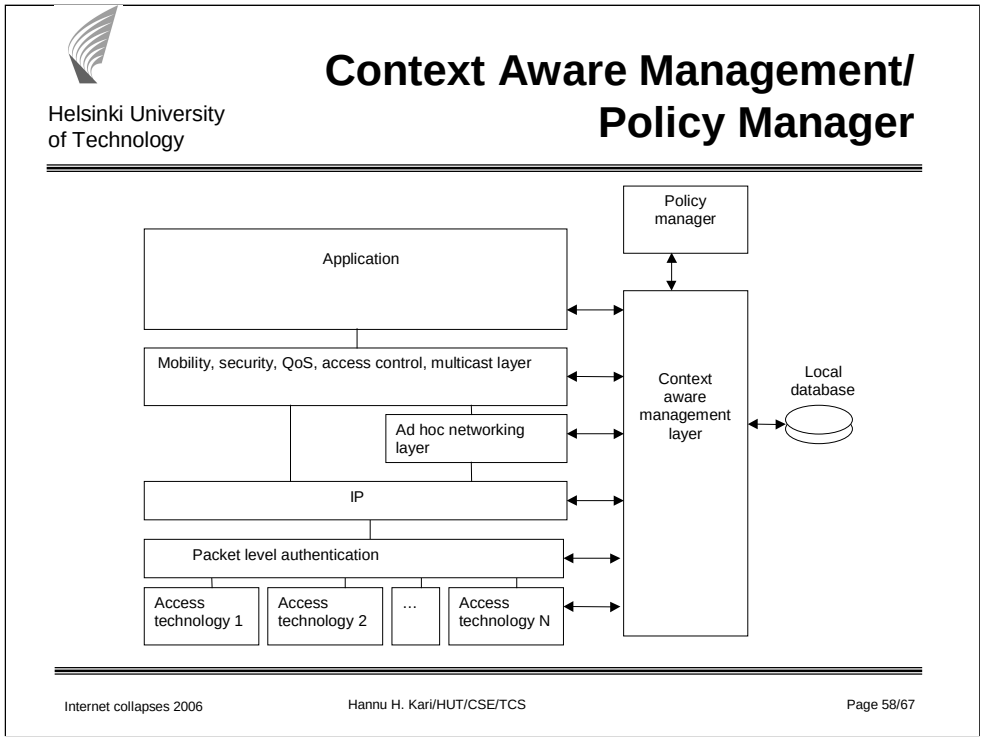
-Don’t trust any more node E1, E2, ..., since their are now controlled by the enemy

-Here are the new rules to nodes E1, E2, ...

The first message ensures that we stop all communication with the compromised nodes. While the second message allows us to se new rules to the compromised nodes.



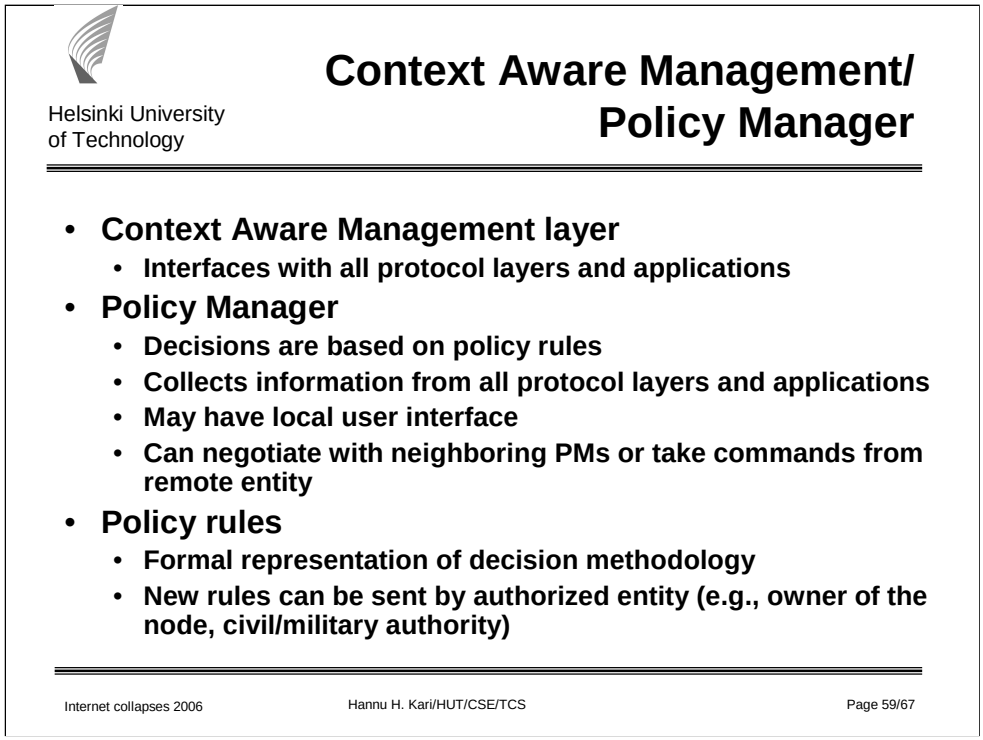
The new rules can guide the compromised nodes to act against the enemy (e.g., by stop working immediately, feed falsified information to the enemy, or any other means to disturb and mislead the enemy).



The previous example used our Context Aware Management/Policy Manager (CAM/PM) –concept that is shortly introduced in this and the next slides.

The main idea of CAM/PM is to allow local decision making to happen in a one centralized entity in each node. CAM interfaces with all protocol layers gathering information and events from the various protocol layers and applications and forwarding that to policy manager. Then, PM will decide accoding to the rules that are available at the local data base.

The key element of CAM/PM is that we can also change the rules on fly. Just like what happened in the previous example. Since all traffic is always authenticated and encrypted (if necessary) we can send over any kind of network new operation rules to any or all of our nodes.

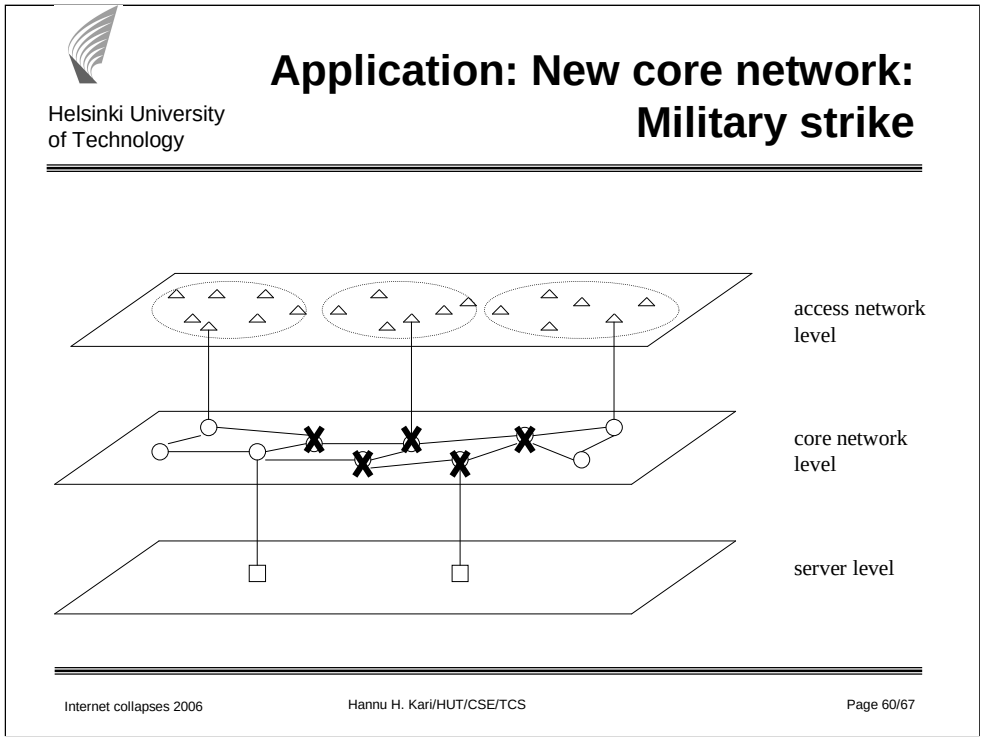


This slide explains more how the CAM/PM is divided into three parts:

-CAM layer that is the interface between starndard protocol layers/applications and our PM

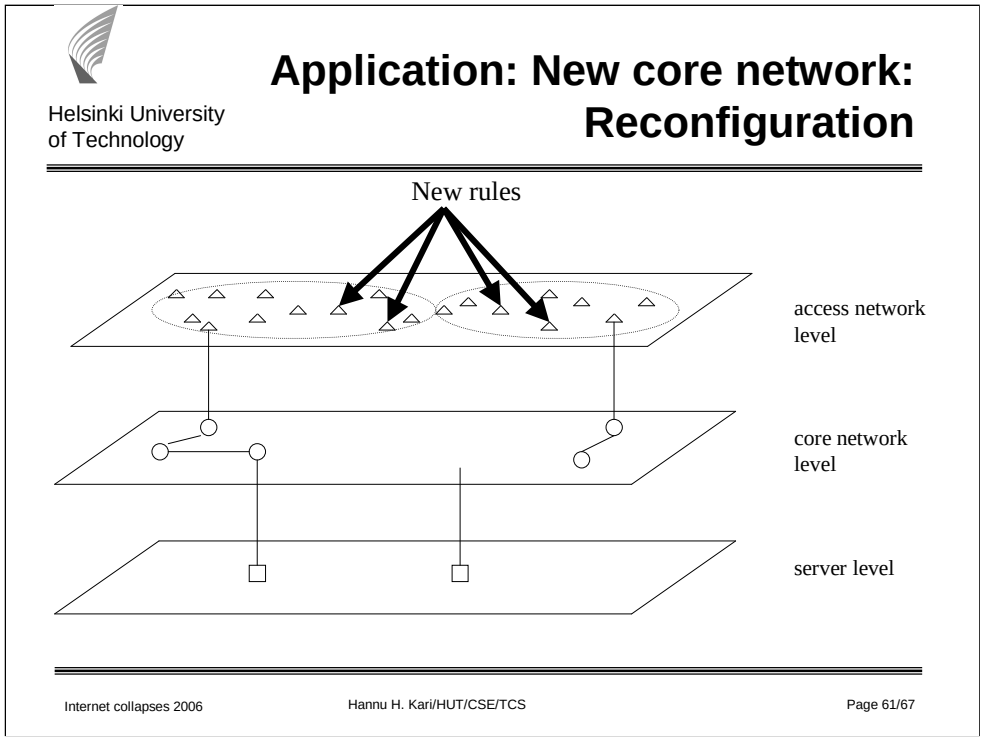
-PM is the main decision engine that has better knowledge than any single application program or protocol layer. Thus PM can see the bigger picture than any individual module.

-We are working on rule based system that allows us to easily update the rules and change the node behavior. This is very useful both in the military but also in the commercial environment. The next example illustrates more...

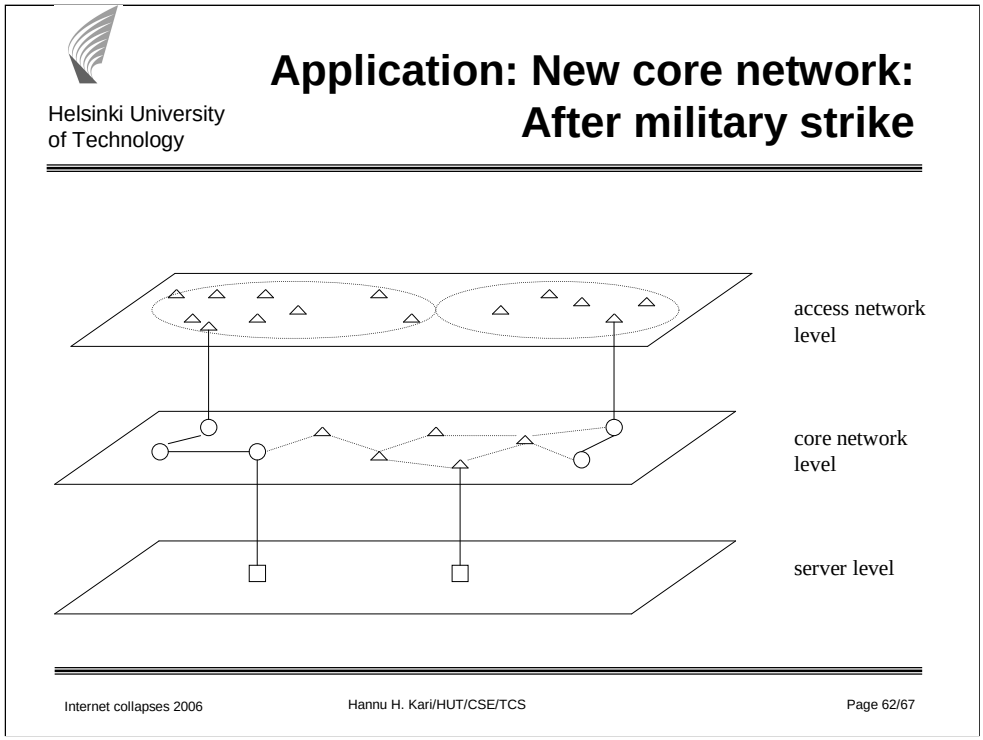


The final example illustrates how the operation of ordinary nodes can be dramatically changed using PLA and CAM/PM.

This example has a military network in which we have three functional levels of nodes (at the same physical area): servers, routers and wireless sensors. In the strategic strike, the enemy has destroyed significant part of our core network thus paralyzing our normal communication.

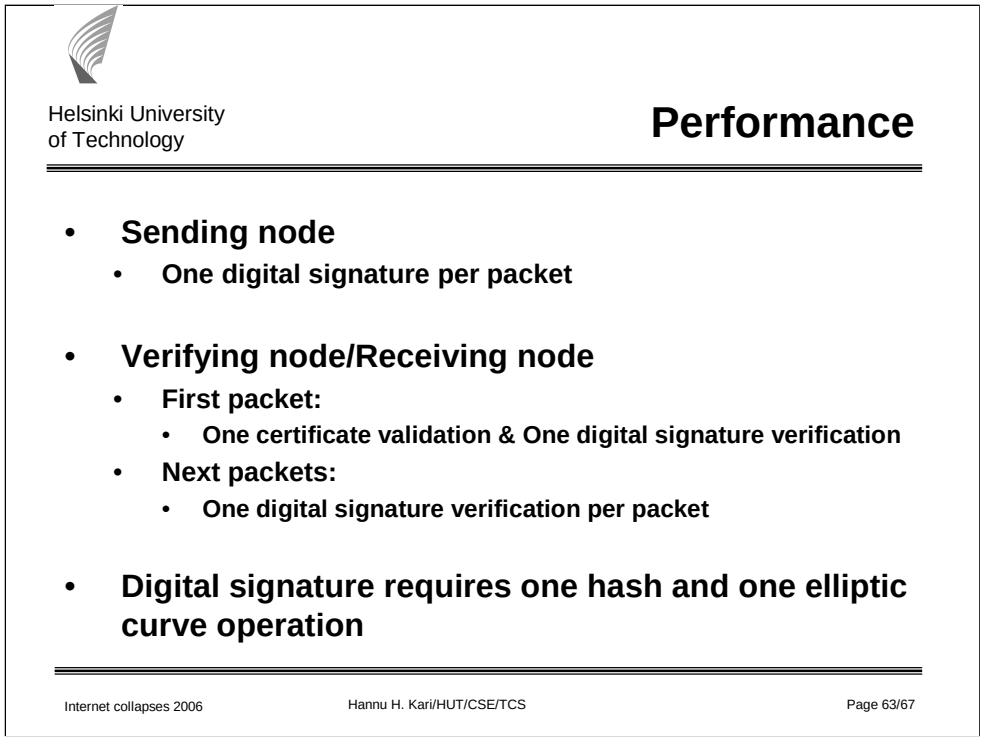


In order to regain our communication, we'll send new rules to some of our sensors: "Forget what ever you were doing and start operating as core network routing nodes regardless of your battery saving rules".



Then those selected sensors will for an ad hoc wireless network, of which only purpose is to route packets as part of the core network. Hence, connectivity is regained.

Obviously, the new core network has significantly less capacity than the old fixed core network, but this will be taken care by CAM/PM also. To the core network and access networks, we need to send also new rules that says: ”Our core network capacity has dropped dramatically, you must prioritize the traffic”.



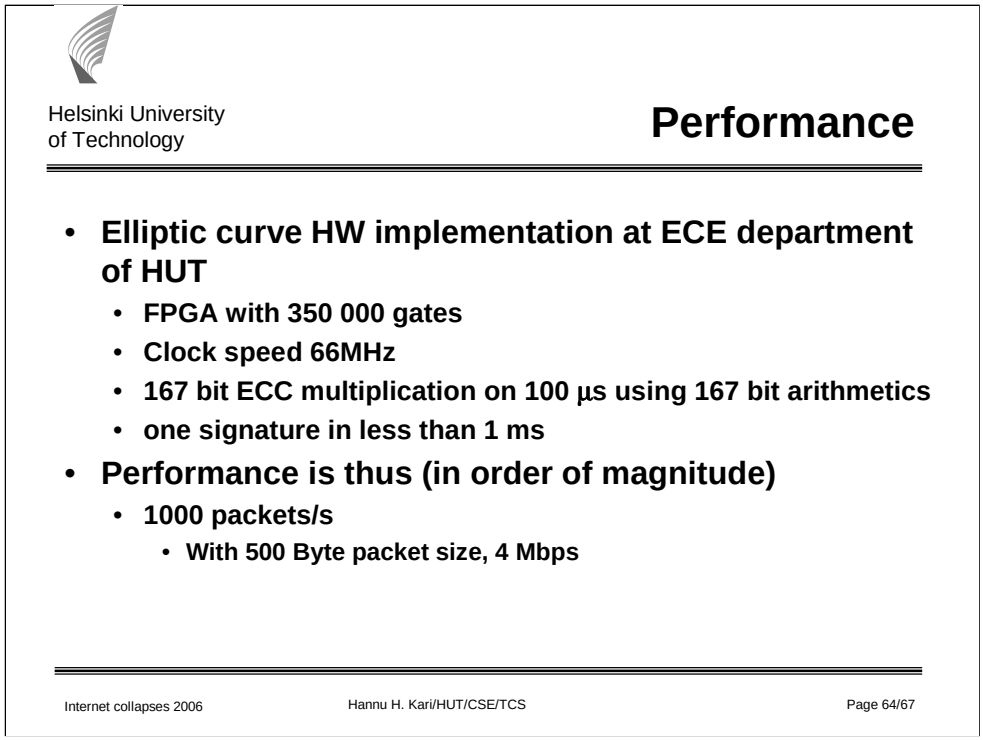
An interesting question on PLA is of course the performance. There are two aspects:

-Overhead caused by PLA. This is at the moment in the order of 100 bytes extra per every packet

-Processing overhead per every packet:

- Sending node needs to sign every packet
- Receiving nodes needs to validate either
  - two signatures (the first packet that it sees from the sender)
  - one signature (for next packets from the same sender, since the certificate validation of the authority can be cached)

This means practically digital signatures one or two per packet!



An example of the speed of digital signatures made in HW is one of our projects at HUT. In GO-SEC –project, we have implemented elliptic curve on FPGA size and speed listed above. With 167 bit keys, the HW is capable of performing about one multiplication in 100 us. This leads roughly about 1 ms time per digital signatures.

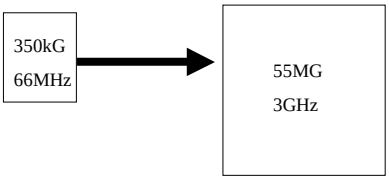
The performance is not much then. Just 1000 signatures per second, that is with 500 byte packet just 4 Mbit/s.

So, we need to do better...

Helsinki University
of Technology

Performance

- **How about scaling up?**
  - **Pentium IV class silicon**
  - **Clock speed**
    - **66MHz -> 3 GHz**
    - **(speedup factor 45)**
  - **Dice size**
    - **350 000 gates -> 55 M gates**
    - **(160 parallel signature units)**



$$\frac{1}{1ms} \times \frac{C_{new}}{C_{ref}} \times \frac{G_{new}}{G_{ref}} = \frac{1}{1ms} \times \frac{3GHz}{66Mhz} \times \frac{55\,000\,000}{350\,000} = 7.14\ Msignature/s$$

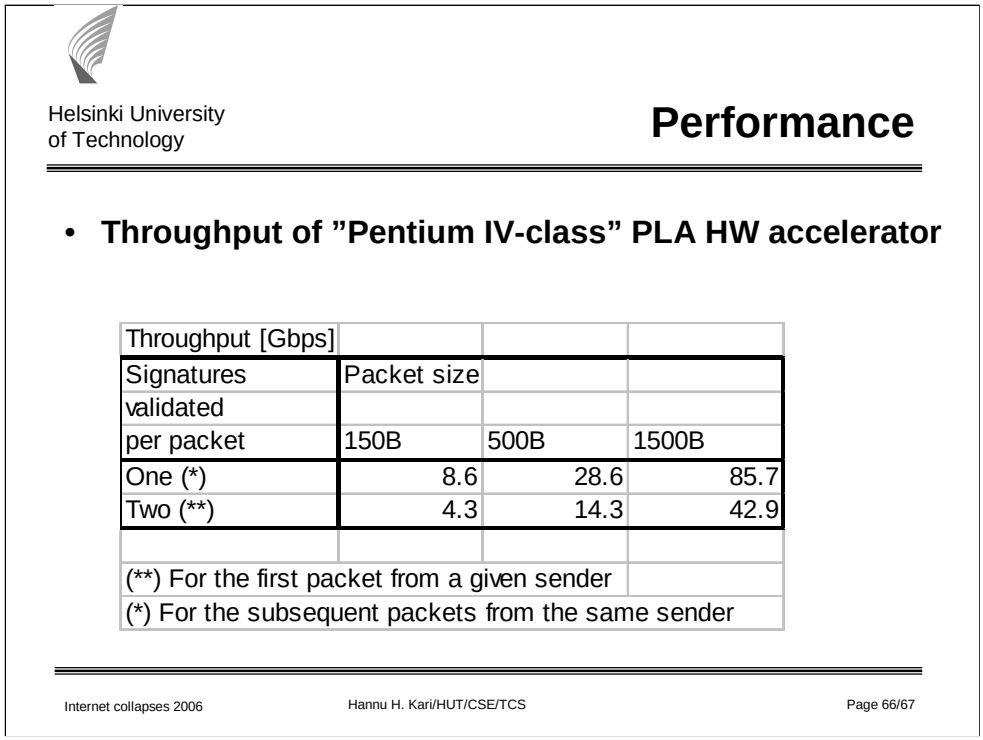
Internet collapses 2006 Hannu H. Kari/HUT/CSE/TCS Page 65/67

What if we scale up the HW? 350 000 gates is not much nowadays, neither is the clock speed of 66 MHz.

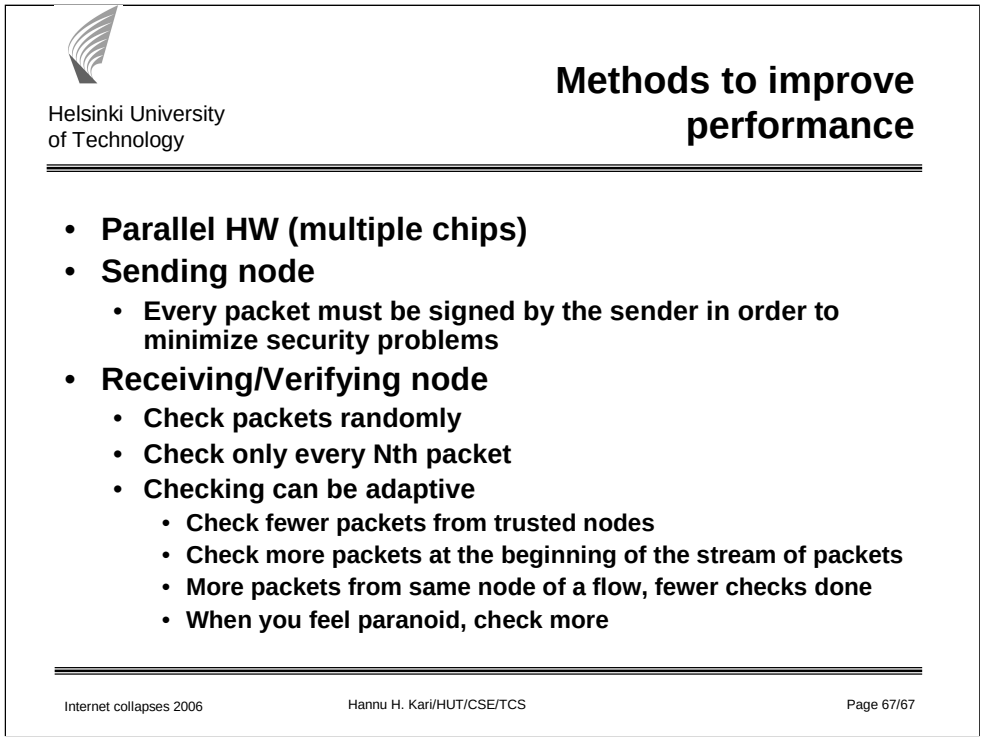
So, let's take a silicon that is the order of magnitude of Pentium IV –processor:
-clock speed of 3 GHz (instead of 66MHz), that leads speed up factor of 45
-gates 55M (instead of 350 k), that leads speed up factor of 160.

Then, we can do rougly 7 Million digital signatures per second.

PLA is very well scaleable by the clock speed and also chip size. We can put several parallel PLA verification units in processing different IP packets in parallel.



Hence, the thourghput estimation what we are assuming to gain with reasonable sized specialized HW chips is in the order of gigabits per second. In practice, we can say that we can take standard 10 gigabit Ethernet in (with wirespeed) and verify all the packets, and pass only the valid ones.



How we could accelerate this even further (if we would need to handle 100 GE Ethernet interfaces)?

Obvious solution would be to have several parallel chips

In the sending side, we need to sign every packet in order make sure that none of the packets can be modified on the path.

But at the verifying side, we can do lots of optimization:

-We don't need to check every packet in every router, just check them randomly and assume that other routers are doing the same. Once verification fails, report to upstream and we can check every packet.

-We may also do checking adaptively. First we check more frequently, but once the stream is going fine, then we do less checkings.