The Secure Access Node Project:
A Hardware-Based Large-Scale Security Solution for Access Networks

Jens Rohrbeck, Vlado Altmann, Stefan Pfeiffer,
Peter Danielis, Jan Skodzik, Dirk Timmermann
University of Rostock
Institute of Applied Microelectronics and Computer Engineering
Rostock, Germany
{jens.rohrbeck;dirk.timmermann}@uni-rostock.de

Matthias Ninnemann, Maik Rönnau
Nokia Siemens Networks GmbH & Co. KG,
Broadband Access Division
Greifswald, Germany
{matthias.ninnemann;maik.ronnau}@nsn.com

Abstract—**Providing network security is one of the most important tasks in todays Internet. Unfortunately, many users are not able to protect themselves and their networks. Therefore, a novel security concept is presented to protect users by providing security measures at the Internet Service Provider level. Already now, Internet Service Providers are using different security measures, e.g., Virtual Local Area Network tags, MAC limitation, or MAC address translation. The presented approach extends these security measures by three hardware-based security subsystems. A firewall engine controls the header of Ethernet frames, Internet packets, and the next following protocols. Furthermore, a Web filter module disables access to violent and child pornography Web content. The third subsystem is a Bloom filter-based deep packet inspection engine to observe the payload after the protocol header. Based on deep packet inspection, it is possible to detect network intruder. A firewall, a Web filter as well as a network intrusion detection system, at the ingress of the network, offer security measures to all connected users, especially to users with limited IT expert knowledge. Each of the mentioned systems has a powerful packet classification engine and a high speed rule set engine used by the firewall to find specific rules for each frame. The rule set engine does not need expensive content addressable memory. All described filter modules as well as the packet classification engine and the rule set engine are developed in reconfigurable hardware. Thus, rule updates and adjustments to the hardware are easy to realize. Adjustments can be made only by the Internet Service Provider administrator. Consequently, the security system itself is secured against attacks from users and from the network side. This novel security approach allows for the protection of up to 32,000 Internet users in wire speed. Furthermore, the prototype system is able to process network traffic at wire speed.**

Keywords-Internet Security; Access Network; Hardware Firewall; Hardware Web Filter; Hardware Intrusion Detection.

I. Introduction

Firewalls and anti-virus programs provide basic protection for Internet-enabled devices. Normally, these security measures are installed on computers of users. But installing security measures at the users' side has two serious drawbacks. Firstly, the threat detection is done on the target machine. Secondly, the users must install, upgrade, and maintain these security measures without professional support. Other measures such as a Web filter and a deep packet inspection engine like snort are often not installed and require additional maintenance. In addition, the majority of Internet users is missing the necessary expertise to configure their security software so that it provides optimal protection. Furthermore, because of negative experiences like phishing attacks targeting online banking, many users have lost their confidence in online services and the Internet itself. Therefore, it is mandatory to disburden respectively to support users in issues of Internet security.

A trustworthy place for the placement of security measures is the ingress of the network — the access network. Each user, referred to as subscriber by Internet Service Providers (ISPs), is connected to the Internet through the access network. The access network itself consists of access nodes (AN). As ANs are transparent for subscribers, these components are safe from, e.g., Denial of Service Attacks. To reestablish the subscribers' confidence into the Internet and moreover, to even protect the Internet itself, it is useful to establish additional security services at ANs. With these additional security features, two objectives can be achieved. On the one hand, the subscriber is offered a higher security service without the need to care about security measures himself. On the other hand, outgoing traffic from subscribers can be verified. Thus, the network is protected as well.

Additionally, if the subscribers and the global network are protected, services inside access network like Dynamic Host Configuration Protocol (DHCP) and Domain Name Service (DNS) are protected as well. Although an antivirus program is mandatory to protect a network, the presented system excludes this protection measure. Firstly, antivirus programs are for free. They can offer a good protection with default settings and the maintenance of such programs is very simple because they update themselves autonomously. Secondly, the resources to monitor antivirus signatures and other maleware signatures exceed the available resources of an AN. The access network, as the ingress of the network, aggregates the traffic of many single Internet connections. So, traffic rates from some Mbit/s up to more than 100 Gbit/s have to be processed — both in up- and downstream.

Furthermore, the goal is to support rules for up to 32,000 connections. Due to hardware restrictions, a renouncement of connection tracking and the control of protocols' communication sequences is necessary. The prototype referred to as Secure Access Node (SecAN) presented in [1] extends the currently available security measures on an AN by a packet filtering firewall, Web filtering, and intrusion detection system (IDS). Thereby, this functionality moves from the subscriber to the ISP.

To fulfill these tasks under the conditions described, a very powerful packet classification [2] and packet processing are required. Due to these requirements, pure software solutions are not applicable. Therefore, SecAN is a hardware solution on a XILINX evaluation board with a FX70T Field Programmable Gate Array (FPGA). This solution do not use CAM memory. Already for 224 connections (these approximates ca. 0.7 % of all connections), over 90 % of available block ram ressources or 23 % of slice register would be needed. Without using CAM, the solution is able to control traffic at wire speed.

Through the development and deployment of the security measures, the operators will have financial effort. But at the same time, the ISP will have a benefit against ISP which do not use this security solution. He can take up new security measures in its portfolio and so he can offer network protection for subscribers without the need to care about it. This creates a new service, whereby the funding is guaranteed through subscribers.

Briefly summarized, the main contributions of this paper are the following:

- The presented prototype is a novel hardware solution of a packet filter, Web filter, and an intrusion detection system placed onto an access network. All filter subsystems can be used independently or together as a closed system.
- This solution is able to control traffic in up- and downstream direction simultaneously without packet loss. Thus, it can protect the connected subscribers and the network itself.
- SecAN uses a set of 10 frame parameter to classify connections individually at which the number of configurable rules is not limited as described in [2].
- As target platform, a XILINX evaluation board with an FX70T FPGA is used. Although no CAM is used, SecAN is able to control traffic in wire speed, at least with 1 GBit/s. Because the speed is module dependent the system is able to achieve up to 4.8 GBit/s.
- The structure and functionality of all developed modules as well as the used resources and the reached speed are described in detail.

The remainder of this paper is organized as follows: Section II describes security measures available in the access network today. In Section III, the SecAN's hardware architecture is presented. Here the various modules and their functions are explained. Following, in Section IV, the generation of different clocking domains is described. Before the paper concludes in Section VI, the developed software solution for flexible configuration of the hardware is presented in Section V.

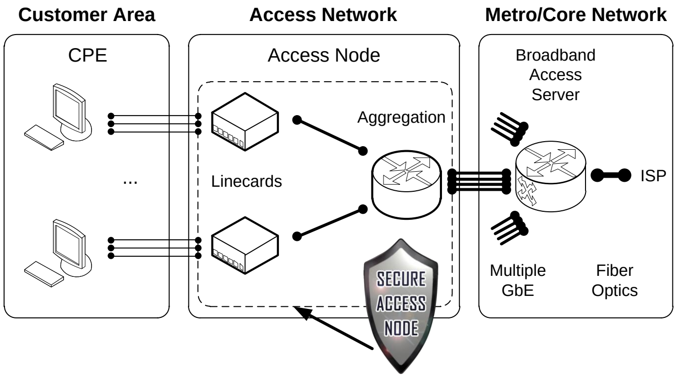

Figure 1. Access Network containing ANs. The Secure Access Node is an extension of an AN.

II. Security Measures In The Access Network

Each subscriber achieves access to the Internet through the access network. Access networks comprise subscriber premise equipments (CPEs) and access nodes such as Digital Subscriber Line Access Multiplexers (DSLAMs). The latter usually consists of linecards and aggregation cards as shown in Figure 1. While aggregation cards provide high-bandwidth interfaces towards metro or core networks, linecards aggregate various subscriber lines.

Although the network ingress is transparent for traffic from and to subscribers, ISPs have to protect the access network. Today, security measures mainly include passive measures on OSI layers II and III [3], [4]. For example, ISPs are using security measures like:

- Port isolation - subscriber may not communicate via an AN
- MAC antispoofing - a Source MAC address is allowed only at one port at a time
- MAC address limitation - to limit the number of MAC addresses per port
- MAC address translation - subscribers MAC address is translated to an ISP MAC address
- VLAN tags - to separate subscriber and services
- IP antispoofing - only the IP address - assigned by the ISP - in combination with the requested MAC address, is allowed Source IP and Source MAC pair at a special port

To ensure a minimum necessary level of security when connecting to the Internet, the already introduced security measures must be integrated into the access area by means of the Secure Access Node.

III. SecAN - Architecture

A. Hardware Overview

To emulate the SecAN on an AN, a XILINX ML507 evaluation board with an FX70T FPGA [5] is utilized. Thereby the FPGA is the main component and is typically used on linecards. Furthermore, the 1 MB Static Random Access Memory (SRAM) and the 512 MB large Double Data Rate Synchronous Dynamic Random Access Memory (DDR2-SDRAM) is utilized. To control traffic in upstream and downstream direction, two 1 Gigabit Ethernet transceivers are used as well.

B. The System In General

Each Ethernet transceiver of the evaluation board is able to process data with 1 Gbit/s. That corresponds 16 bits per clock cycle which have to be processed. To avoid the discarding of any uncontrolled data frame and due to the internal delay during frame processing, an increasing of the internal bandwidth to 32 Bit/cycle is necessary.

Generally, the SecAN system is divided into two hardware groups. The inner hardware group is the actual filter core and consists of packet classification engine (PCE), rule set engine (RSE), and packet processing engine (PPE) and is used for processing of network traffic. Furthermore, the outer hardware group consists of two Ethernet transceiver, a frame multiplexer, a frame demultiplexer, and a configurator and is used for receiving and sending Ethernet data as well as the receiving of configuration information. These components consume the resources and achieve the speed as shown in Table I.

Modul	Flip Flop / LUT Slices	BRAM	Speed
SecAN firewall	1993/1921 (≙5 %/(≙5 %)	8 (≙6 %)	173.273 MHz (≙5.54 Gbit/s)

Table I
RESOURCES AND SPEED FOR ALL SECAN HARDWARE MODULES WHICH RECEIVE AND SEND EXTERNAL DATA

The used resources are very low. Thus many resources remain for the main task - the packet processing. Nevertheless, the outer hardware group limits the maximum attainable speed to 5.5 GBit/s.

C. Configuration and Frame Processing In General

- Before the system can process traffic, it must be configured. The components that need to be configured are the PCE, RSE, Web filter, and the DPI control stage. All configuration data is solely written to the hardware and read from it by the ISP. The configuration flow is shown by dashed arrows in Figure 2.
- After configuration, frames reach the inner system. The frame multiplexer receives and buffers Ethernet frames from the evaluation board's interfaces and chooses the next frame to be processed by the PCE. The PCE separates flow data from the frame and requests the particular rule set from the RSE. Each rule set is a particular collection of rules, which are necessary to evaluate a frame. After identifying the right rule set, it has to be forwarded to the PCE. When the rule set reaches the PCE, the rule set, the data frame, and collected frame parameters have to be sent in the direction of the PPE - to the control stages. In the control stages (CS), the rules from the rule set are applied. The CSs are able to discard or forward frames or replace frame values like IP addresses. If a frame is not discarded it leaves the PPE towards to the frame demultiplexer. The frame demultiplexer discards the rest of the rule set and the frame parameter set and forwards the frame to the right output interface.

D. Hardware Configuration

An initial configuration as well as an on-the-fly configuration can be done but before the hardware components are not configured, no frames traverse the SecAN. The configurator module, shown in Figure 2, receives the writing configuration data from the configuration software. For control reasons, configuration data can be read via the configurator.

During configuration, no new frames can be process. For that reason, two possibilities can be applied. Either, all data traverses the system uncontrolled or all data is blocked. Because a security system should not be bypassed, it is better to block all data during the configuration phase.

The configuration starts with a 'start of configuration' information. All internal processes are stopped and all new frames are rejected. Following, the configuration data are received and distributed by the configurator. The last configuration data is a 'end of configuration' information. After that, the configuration process is finished and data frames can be processed by the SecAN.

Configuration data is provided to the appropriate modules by the configurator. This data has a type-length-value layout.

- Type is an 8 bit field and determines the component of the hardware to be addressed. Each component has two valid type values: one for writing configuration data and one for reading configured data.
- Length is an 8 bit field and represents the number of configuration data bytes. A maximum of 256 bytes plus configuration header can be configured.
- The actual configuration data is contained in the value part. All components are assigned specific configuration values.

Resources and Speed: Table II shows the required resources as well as the speed of the configurator module based on an Virtex 5 FX70T FPGA. Just as the modules

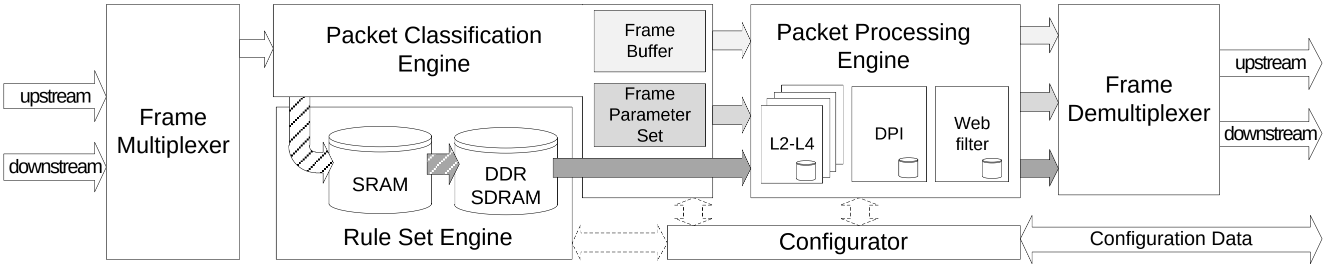

Figure 2. Block diagram of the Secure Access Node

for receiving and sending data the configurator modul is not among the core components. The hardware requirements are very low for this module. However, the maximum processing speed cannot be achieved because the transceiver modules limit the processing speed.

Modul	Flip Flop / LUT Slices	BRAM	Speed
Configurator	851/1300 (≙2 %/≙3 %)	1 (≙1 %)	195.236 MHz (≙6.25 Gbit/s)

Table II
RESOURCES AND SPEED FOR THE CONFIGURATOR

E. The Frame Processing Flow

The frame processing flow is shown by shaded arrows in Figure 2. If the PCE is not busy it has to receive and classify the frame. The frame multiplexer selects a frame from the internal buffer with the highest fill level. After frame classification is finished, the RSE searches for an individual rule set for each frame. Rules of the rule set are applied to the frame. If the frame is not discarded by the PPE due to the rules it is sent to the correct output interface by the frame demultiplexer.

1) Packet Classification Engine: The PCE fulfills several tasks. Briefly summarized, it buffers the whole frame and creates an special parameter set. In this parameter set, different frame parameter and special information for the Web filter and the IDS control stage are saved. Later, all control stages use this special parameter set to accelerate the decision whether the frame is accepted or dropped. Furthermore, it separates a distinct key from the frame - the Flow ID. The creation of the Flow ID depends on the receiving direction of the frame. Therefore, the PCE has got additional information from the frame multiplexer. This flow ID is the basis for a connection-specific rule set. Moreover, for a quick search for the rule set, the PCE determines the memory address.

Creation of the Flow IDs: During the configuration, the PCE has got two so called Flow ID trigger. These triggers describe, which of the frame parameters are necessary to classify a frame. It is possible to set a new trigger by reconfiguration on the fly. One trigger is used for upstream frames and the other trigger for downstream frames. With one of these trigger in combination with the received information of the frame multiplexer, the PCE is able to create a directional unique Flow ID. Additionally, a Flow ID valid flag signals the validation of the unique Flow ID. That means, if at least one of the necessary frame parameter is not be available the Flow ID is invalid. As a result, a default rule set must be ordered. Often packets are classified by five packet header fields: Both IP addresses and port numbers, and transport layer protocol [6]–[8]. To achieve a higher degree of flexibility during frame classification both MAC addresses, up to 2 VLAN tags, and the Ether type field are added to the frame parameter set.

Calculation of a memory address: Similar projects like [6]–[8] have very short Flow IDs and use CAM or bloom filter approaches to increase the lookup performance as suggested in [2]. So, they do not need address calculation for memories. If CAM is used this either results in high acquisition costs or a disproportionately high consumption of hardware resources. Furthermore, bloom filter approaches are not able to calculate a memory address for a rule set. Other hardware friendly approaches are tree based. Although a logarithmic time complexity promises good results, but a quicker working solution is needed. Therefore, a hardware-friendly CRC hash algorithm is used. This kind of algorithms compresses an input vector to an output vector, at which the length of the input vector is usually higher than the length of the output vector. In this case, the output vector - the hash value - is used as memory address. Although CRC is not a perfect hash function, the big advantage is that CRC provides uniformly distributed binary vectors. The aspect of collision resolution is discussed in Section III-E2.

After the Flow ID is completely composed and the hash value is calculated, a request for the rule set is performed by the RSE. Together with the Flow ID, the PCE has been set a ”Flow ID valid” flag. The RSE decides with the help of the ”Flow ID valid” flag, which rule set should be requested - the individual rule set from the calculated address or the default rule set. If the individual rule set is received from the RSE, the frame, rule set, and parameter set is sent towards the PPE. Because only the parameter set is available in the

PPE with the first cycle, a comparison with rule parameter can be done before the proper frame data reaches at the PPE.

Generation of log data: Particularly in the access network, hardware solutions are very expensive. Therefore, the optimal use of existing resources is required. Thus, the generation of log data is extremely important. For that reason, the PCE is able to capture log data. Thereby, all frames and all bytes are counted. In combination with the captured log data of the filter modules statistics can be created about the traffic. Thus, new rules can be created and existing rules can be optimized. In this way, existing hardware resources are saved.

Resources and Speed: Table III shows the required resources as well as the speed of the PCE based on an Virtex 5 FX70T FPGA.

Modul	Flip Flop / LUT Slices	BRAM	Speed
Packet Classification Engine	593/865 (≙2 %/≙2 %)	5 (≙4 %)	165.893 MHz (≙5.31 Gbit/s)

Table III
RESOURCES AND SPEED FOR THE PACKET CLASSIFICATION ENGINE

2) Rule Set Engine: For the rule set search, a two-stage approach with a hardware-gentle compression method is used. Firstly, the mapping between Flow ID and the rule set is done in a sufficiently large SRAM memory. Two clock cycles after the application of an address to the SRAM, the date is available. Secondly, very large rule sets have to be stored in DDR2-SDRAM. To increase speed when reading and writing memory information, the given memory controller [9], which does not use the evaluation board's internal bus systems, has been extended. The rule set is sent towards the PCE and forwarded together with the frame and frame parameter set to the PPE.

Flow ID Mapping in SRAM: When the packet classification is finished, the RSE gets the Flow ID, the CRC hash value of the Flow ID, and the validation information of the Flow ID from the PCE. Should the "Flow ID valid" bit indicate an invalid Flow ID, the memory address for the default rule set is selected and a request to the DDR2-SDRAM is sent. Normally, a specific SRAM entry has to be searched. As described before, CRC is not a perfect hash function and so the aspect of collision resolution is extremely important. For this reason we have developed two strategies. Firstly, a linear collision resolution has been tested. Therefore, thirty runs have been made, each with 32,000 valid Flow IDs and we have calculated the CRC hash values of all Flow IDs and ordered it after collisions. In the 1 MB of SRAM, 43,690 of the linking entries can be saved. All Flow IDs, which do not cause collisions, are stored. After that, all entries, which cause one collision, are stored as well. Whenever a memory location is found, which is occupied, a linear search for a free memory location begins.

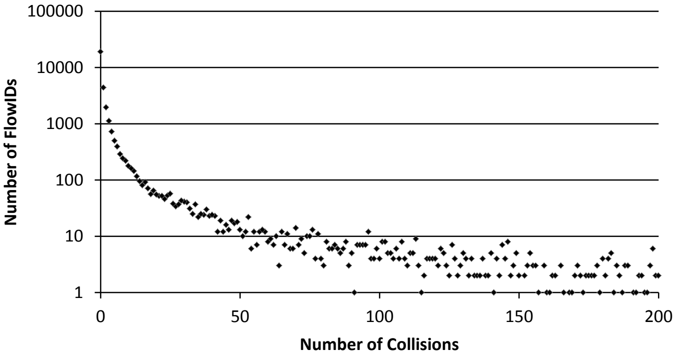

Figure 3. Linear Collision Resolution for Flow IDs. For the y-Axis, a Base 10 Logarithmic Scale is Used.

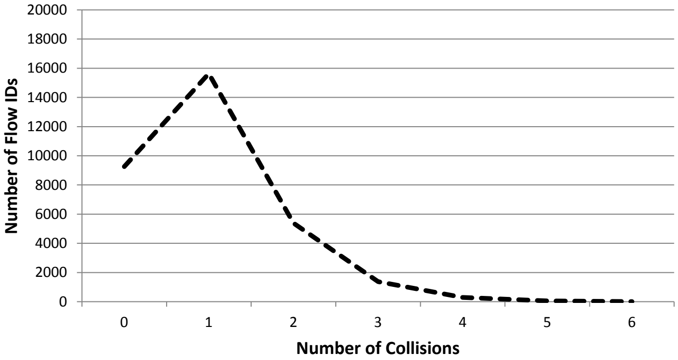

Figure 4. Indexed Collision Resolution for Flow IDs.

Although the SRAM is large enough, in the worst case, up to 1617 collisions are reached (as shown in Figure 3). This solution undoes the benefits of CRC.

The second strategy uses indexed memory entries. That means, after all entries without collision have been saved, all remaining entries are stored in free memory slots. Now, each memory entry contains a collision information and a SRAM address information. If the collision information indicates another entry on the same SRAM position the following SRAM address information is used. This procedure is repeated until the correct entry is found. If the desired Flow ID has not been found but the collision information indicates no further collision, the address of the default rule set is used. This case should not occur as all of the correctly collected Flow IDs are known in advance.

In approximately 29 % of all test cases, the calculated CRC value corresponds directly with the searched memory address and in about another 49 % the searched address is achieved after one collision. In the worst case, 6 collisions occur. The result is significantly better than a logarithmic search tree, which needs up to 16 steps to search all 43,690 possible memory entries. Figure 4 shows the test results.

Rule Set Order in DDR2-SDRAM: The DDR2 memory stores all available rule sets. A rule set is a collection of individual rules, which are used by the firewall filter stages.

By definition, a rule set has a maximum length of up to 1,024 bytes. Thus, up to 262,144 rule sets can be stored in the available DDR2 memory. After the DDR2 controller has got the DDR2 address information from the SRAM controller, approximately 27 clock cycles have to be waited until the first data arrives at the DDR controller. The rule set is preceded by a head, sent towards the PCE, and forwarded together with the frame and frame parameter set to the PPE. The self-developed SRAM controller and the extended DDR controller as well as increasing the internal bandwidth will guarantee maximum throughput for the entire SecAN system. Table IV shows the required resources as well as the speed for the SRAM controller module and the DDR2 controller modul based on an Virtex 5 FX70T FPGA.

Modul	Flip Flop / LUT Slices	BRAM	Speed
SRAM controller	470/1048 (≙2 %/≙3 %)	6 (≙4 %)	179.727 MHz (≙5.75 Gbit/s)
DDR2 controller	2807/1842 (≙6 %/≙4 %)	6 (≙4 %)	179.340 MHz (≙5.74 Gbit/s)

Table IV
RESOURCES AND SPEED FOR THE SRAM CONTROLLER AND DDR2 CONTROLLER

3) Packet Processing Engine: The PPE is responsible the for control and evaluation of the data stream and consists of three central components. In addition to a classic packet filtering, a Web filter and a signature recognition engine have been implemented. Each of the three components aims at protecting subscribers from unauthorized access from the network side and suppresses attacks from subscribers on the network.

Packet Filtering: The packet filter is divided into 12 control stages, so each CS has only a marginal role to fulfill. On OSI layer 2, a source MAC and a destination MAC CS has been developed as well as separate CS's for both possible VLAN tags and the ethertype. Furthermore, to control IPv4 parameter, a CS for the OSI layer 4 protocol and 2 separate CS's for both IP addresses have been designed. Moreover, 2 CSs control OSI layer 4 port information. Last but not least, a MAC address translation as well as IP antispoofing CS has been developed. Due to the modularity of the PPE, the whole system is very flexible and efficiently to extend. Figure 5 shows on the one hand the outer structure of all CS's and on the other hand the design structure for the OSI layer 4 port CS's.

The design of Figure 5 is separated into two parts. In the upper part, the Ethernet frame is processed and in the lower part the logic for the log data is shown.

The Ethernet frames pass all CS's one after the other. Simultaneously with a frame, the belonging parameter set as well as the rule set reaches at the CS. Rule sets can have one or more rules whereby each rule has a type-length-value composition similar to configuration data. To speed up the processing, the rules in the rule set have been configured in the same order as the CS's are arranged. Because each applied rule is removed, each CS has to look only at the first rule. Moreover, each CS has a unique identifier. If the type of the first rule does not equal the CS's ID, the frame, rule set, and parameter set are forwarded to the next stage. Otherwise, the rule is processed by the CS. Each CS compares the data from the rule with the data of the parameter set. Because the whole parameter set is available in the first cycle of a new frame, the lookup increases the processing speed, especially for OSI layer 4 values. In case of a match, the rule action has to be executed. That is, the frame, rule set, and parameter set can be discarded or forwarded as well as frame parameter can be changed. After processing, the applied rule will be removed and the next CS is able to look at the first position of the rule set. According to the principle of divide and conquer, the rule set finit state machine (FSM) and the frame FSM receives and buffers incoming data. The port FSM analyzes the rule set, applies the rule and forwards or discards the buffered data.

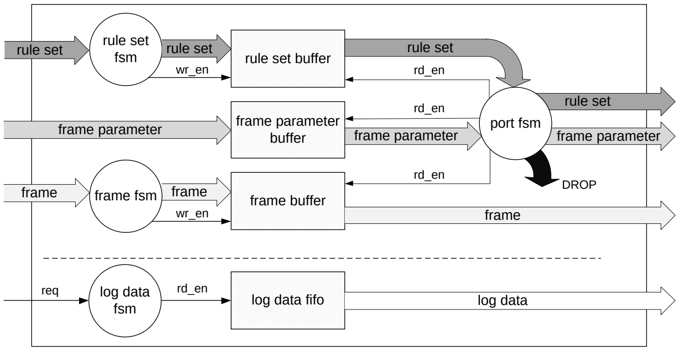

Figure 5. Outer and inner design of the port control stages

If the rule action requires the discarding of the received data, there are two counter values, which have to be increased. The drop frame counter counts the discarded Ethernet frames and the drop byte counter counts all discarded bytes. Both values are stored in a 32 bit register. Additionally, the reason for discarding is stored in a FIFO buffer, e.g., the existing and the required MAC addresses. If the buffer reaches the maximum fill level, the oldest date is replaced by the current. Prospectively, a log data collector will request the stored data from all CS's. The log data FSM will get a request signal and send all captured log data in the direction of the log data collector.

Table V shows the required resources as well as the speed for all described CSs based on an Virtex 5 FX70T FPGA.

Throughput and Resources of the SecAN Subsystems: The SecAN project consists of three subsystems. The SecAN packet filter firewall subsystem, consists of two Ethernet interfaces, two receiving and sending synchronization frame buffers, a frame multiplexer and a frame demultiplexer, the

Modul	Flip Flop / LUT Slices	BRAM	Speed
Source MAC and Destination MAC CS	701/657 (≙2 %/≙2 %)	1 (≙1 %)	173.322 MHz (≙5.55 Gbit/s)
Inner VLAN and Outer VLAN CS	623/689 (≙2 %/≙2 %)	1 (≙1 %)	176.444 MHz (≙5.65 Gbit/s)
Ethertype CS	688/710 (≙2 %/≙2 %)	1 (≙1 %)	180,101 MHz (≙5.67 Gbit/s)
OSI L4 Protocol CS	695/722 (≙2 %/≙2 %)	1 (≙1 %)	179.556 MHz (≙5.75 Gbit/s)
Source IP and Destination IP CS	699/695 (≙2 %/≙2 %)	1 (≙1 %)	172.311 MHz (≙5.51 Gbit/s)
OSI L4 Source and Destination Port CS	593/865 (≙2 %/≙2 %)	5 (≙3 %)	182.815 MHz (≙5.85 Gbit/s)
IP Antispoofing CS	675/665 (≙2 %/≙2 %)	1 (≙0 1%)	184.101 MHz (≙5.89 Gbit/s)
MAC Address Translation CS	723/625 (≙2 %/≙2 %)	1 (≙1 %)	178.230 MHz (≙5.70 Gbit/s)

Table V
RESOURCES AND SPEED FOR ALL DESCRIBED CONTROL STAGES

configurator as well as the packet classification engine, the rule set engine and the described 12 control stages. Required resources and reached speed are shown in Table VI. Thereby, the speed is sufficient to process traffic in wire speed so that no packets have to be dropped. In order to avoid packet loss, the overall delay time must be less than or equal to a time needed for the internal processing of 248 bytes. Consequently, the following formula must be satisfied:

$$\frac{D_i}{T_i} \leq \frac{D_e}{T_e}, \tag{1}$$

where D_i is internal data volume, T_i is internal throughput, D_e is external data volume and T_e is external throughput. As the internal throughput is 4.8 Gbit/s and the external throughput is 2 Gbit/s, the formula is satisfied.

Subsystem	Slices Flip Flop pairs	BRAM	Speed
SecAN firewall	14.267/18.456 (≙ 31.9%/≙ 41.2%)	45 (≙ 30.4%)	150.3 MHz (≙4.8 Gbit/s)

Table VI
RESOURCES AND SPEED FOR THE SECAN FIREWALL SUBSYSTEM

Web Filtering: Web filters are a very sensitive issue and have been poorly discussed in the research community. Some countries such as China, the United States, and Great Britain [10] already use Web filtering. The British system ”Cleanfeed” has a two stage structure [10]. In the first stage, the system filters IP addresses. If the IP address matches a request is sent to the external data base to verify the domain. The data base is managed by the Internet Watch Foundation (IWF), which collects reports about criminal online content. ”Cleanfeed” grants an efficient domain filtering. However, it suffers from high latency due to its structure, which constrains the Web surfing experience of users. The US Web filtering system achieves better latency. However, overblocking was substantiated, i.e., the Web sites were blocked although they were not blacklisted [11]. The China Internet filtering system inspects Web traffic for specified keywords [12]. If the keyword is found the Web filter resets the connection by setting the TCP reset flag. The frame that contains the keyword is still forwarded to the recipient. If the endpoints ignore the reset flag the connection persists. Thus, the content can be transported to the requester.

The developed Web filter avoids the drawbacks of the mentioned Web filtering systems. The suggested solution solely utilizes local resources ensuring high processing speed. Moreover, it cannot produce false positives as each domain is exactly verified in the blacklist. Thereby, overblocking is avoided. The packet with malicious content is dropped and though the communication is interrupted.

SecAN Web filter module filters HTTP traffic. It inspects HTTP-GET requests for domain name of the Web server. Afterwards, the domain name is checked in the local blacklist. HTTP-based filtering as opposed to DNS filtering grants immediate effect, which is an essential issue in order to block a malicious Web content. Moreover, HTTP requests to proxies are checked as well. As the authors’ goal is to monitor Web traffic, other protocols should not be blocked. By using HTTP monitoring, the Web filter cannot be simply bypassed by adding the IP address of the Web server into the local hosts file.

High throughput requires fast search algorithms. Therefore, the suggested Web filtering approach has two level search architecture. In the first level search, the domain name is hashed with CRC64 hash function. The calculated hash value is checked in the hash table, which is stored in a cache memory. The rree structure of the hash table grants logarithmic complexity. Moreover, cache access time ensures high search speed. As hashing produces false positives, a second level search is required. The full domain names is stored in the on-board DDR2 SDRAM. If the first level search was successful, start search address for the DDR2 SDRAM is provided. The blacklist has a bucket structure, i.e., all domain names, which generate the same hash value belong to one bucket. The domain names in one bucket are linearly searched. If the requested domain name matches in the blacklist, the HTTP-GET frame is dropped. Otherwise, it is passed through. The described structure is depicted in Figure 6.

In order to test the Web filter module, a data bank with real world domains provided from domain name registrar VeriSign is used [13]. 23 million domains were hashed with the CRC64 hash function and thereby 159 collisions were detected with a maximum of two domains per collision. The collision ratio is $6.9 \cdot 10^{-6}$. As a result, one bucket would normally have only one domain and thus only one DDR2 SDRAM access is required in the second level searching. According to that, the possibility to get a false positive in

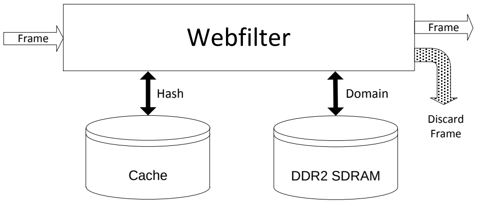

Figure 6. Webfilter structure

the first level searching is under 1‰.

Resource consumption of the Web filter module for the test platform (XILINX FX70T) is depicted in Table VII. The blacklist size was limited to 4096 entries. Growing blacklists result in increasing BRAM and DDR2 SDRAM consumption. However, slice consumption remains constantly low.

Modul	Flip Flop / LUT Slices	BRAM	SDRAM	Speed
Webfilter	897/2319 (≙2 %/≙5 %)	13 (≙9 %)	1.1 MB (≙0.2 %)	112 MHz (≙3.58 Gbit/s)

Table VII
WEB FILTER MODULE RESOURCE CONSUMPTION

The processing of the Web filter's blacklist induce indispensable computation and waiting cycles. In the worst case, 12 cycles cache computation and 54 waiting cycles for DDR2 SDRAM collision resolution are necessary. As the system processes 4 bytes per cycle, the overall delay time corresponds to a time needed for processing 264 bytes. Thereby, the shortest domain name (e.g., *"g.cn"*) can be found after 62 processed bytes. In order to avoid packet loss, the overall delay time must be less than or equal to a time needed for the internal processing of 326 bytes. Consequently, the Formula 1 must be satisfied. In the case of the Web filter, the internal throughput is 3.58 Gbit/s. The external throughput is 1 Gbit/s as stated above. Therefore, a frame has to be at least 92 bytes long to avoid packet loss. However, the length of the shortest possible HTTP-GET frame inclusive interframe gap is 89 bytes. As a 2 KB buffer is used to store incoming frames, packet loss could occur if 683 HTTP-GET requests with minimum length followed each other. This scenario is not realistic on a DSLAM because in practice, the average HTTP-GET frame length is about 400 bytes due to additional HTTP headers. Moreover, HTTP-GET requests represent a fractional part of the overall Internet traffic.

Signature Recognition: To detect malicious signatures at wire speed, Bloom filter-based deep packet inspection technologies are used. The signature detection starts after the header of the transport layer. As there is no clear definition, where to find the signature in the payload, compared with searching for specific header information, signature detection is a problem of massive parallel pattern matching. This problem is solved by concerning pattern matching at wire speed, using a Bloom filter cluster approach (see Figure 7). A Bloom filter is a space-efficient data structure for checking set affinity. The checked element is compressed using several hash functions and depicted to a bitmap-like structure. The hash values serve as bitmap addresses. For programming a Bloom filter, the bitmap is first initialized with zeros. Afterwards, each element of the set is hashed and the bitmap is set to one at the corresponding addresses. For checking set affinity, the alleged element is compressed and looked up in the bitmap. If each address points to a set bitmap element the element is an element of the set with a certain probability. Elements that are recognized as elements of the set but are not due to all their bitmap-elements set by other elements, are called false-positives. The rate of detecting false-positives at a single Bloom filter is called false-positive rate.

For the realization of a Bloom filter based pattern matcher, the Bloom filter set is the signature database whereas the single set element is a specific string. In the implementation, the incoming data stream passes an n byte long monitoring window where each signature length is analyzed by a separate Bloom filter. This is acceptable because every Bloom filter can only hold elements of the same length. Otherwise, the false-positive rate would increase dramatically and no conclusion of set affinity would be possible. The result of the Bloom filter-based analysis is coordinated by an arbiter due to possible simultaneous matches at the same time. One possible algorithm for the analyzer could be longest match first referring to the smaller false-positive-probability for larger signatures. Finally, the match analyzer eliminates all false-positives and generates alarm signals on malicious signature. In this constellation, the Bloom filter cluster plays the role of an optimal pre-filter for the match analyzer reducing the number of possible signature matches found by conventional signature detection algorithms.

As basis for our signatures, the database of the free intrusion detection system SNORT is used, which is parsed and prepared by some external tools for the use in the Secure Access Node. Furthermore, ISP administrator rules are supported, which can be defined in a SNORT-like syntax. To improve the quality of malicious signature detection, additional attack correlated information like protocol type, input port and output port were integrated in the Bloom filter-based pattern matcher to realize a lightweight, hardware-based intrusion detection system working at full wire speed.

The prototype achieves a signature filter-rate of 1.056 GBit/s with a real false-positive rate of less than 0.001 according to the current SNORT database. Building parallel clusters of Bloom filter-based signature matchers, even more throughput can be achieved due to the linear scaling of the filter-rate with the number of instances. The data stream itself is then analyzed by each instance with a particular offset. The resource consumption of the Bloom filter cluster

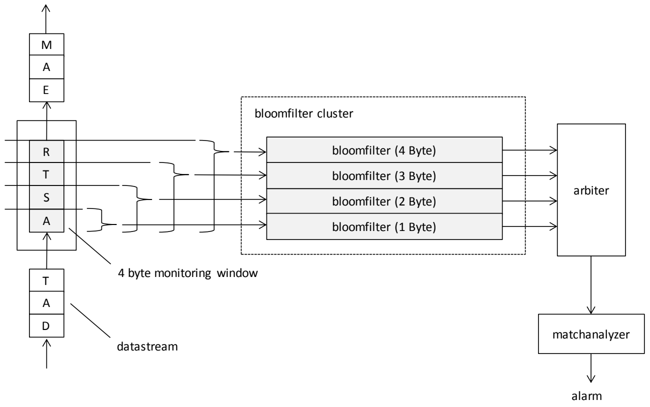

Figure 7. Bloomfilter-based IDS filter cluster for exemplary 4 byte monitoring window

matching signatures with a maximal length of 30 bytes is given in table VIII.

Modul	Flip Flop / LUT Slices	BRAM	Speed
Bloomfilter	16,234/17,150 (≙36 %/≙38 %)	134 (≙90 %)	33 MHz (≙1.06 Gbit/s)

Table VIII
BLOOMFILTER MODULE RESOURCE CONSUMPTION

IV. GENERATION AND DISTRIBUTION OF SECAN'S CLOCK SIGNALS

SecAN has an outer and at least one inner clock domains. The clocks for all clocking domains are generated by clocking moduls using a single digital clock manager (DCM) and global buffers.

In the outer clock domain, the ML507 evaluation board receives and sends data over SecAN's Ethernet interfaces. Both interfaces are able to process data at a speed of 1 Gbit/s. That means, each interface has to process data with 125 MHz and 1 byte per clock cycle. Following the Ethernet interfaces, FIFOs with separated read and write clocks are used to synchronize the outer and the inner clocking domain.

Furthermore, the inner clocking domain depends on SecAN's subsystem: the deep packet inspection, Web filter, and packet filter module. Although the functionality of the DPI module is very complex, the generation of the clock is very easy. Because the DPI subsystem solely uses FPGA internal resources, only one inner clock domain is existent. Hence, the clock for the DPI module as well as the clock for the outer clock domain is generated by the same DCM module.

The generation of the clock for the Web filter subsystem is slightly more complex. Because this control stage uses the DDR2 memory to verify domain matches, there are two inner clock domains - one for the DDR2 memory and one for Web filter's internal logic. The used DDR controller is based on Micron sources. A system internal DCM module uses the 100 MHz board clock as input clock and generates a 200 MHz clock for DDR2 internal processes as well as a 125 MHz clock for all other Web filter modules.

The last of the three subsystems is SecAN's packet filter firewall. This system has the most complex clocking scheme because it uses SRAM as well as DDR2 memory. Both memories have different clocks. Hence, the challenge is to synchronize both memory and the packet filter modules optimally. Figure 8 shows the generation as well as the clock arrangement of the SecAN's firewall subsystem.

Because the packet filter firewall and the Web filter module uses the same DDR controller, the clock generation is exactly identical. DMC 1 generates the 200 MHz and the 125 MHz input clocks of the DDR controller. The controller itself has an internal DCM and generates a new 125 MHz clock. This clock is phase shifted relative to the 125 MHz input clock and should be used by the connected hardware. In case of the packet filter firewall, the SRAM and all firewall internal modules achieves a higher speed of 150.3 MHz. For that reason, the DCM 2 generates from the 125 MHz DDR controller clock three new clock signals. The 125 MHz clock is used for receiving and sending Ethernet data and both 150 MHz clock are used for the internal firewall hardware modules.

Although the SRAM and all firewall internal modules use the same frequency, there is a phase shift between both clocks. The phase shift depends on the distance between FPGA and SRAM memory on the evaluation board. This delay is compensated by a SRAM feedback signal (shown in Figure 8). The phase shift depends on the achievable speed and is 180 °. For example, if the speed is reduced to 80 MHz the phase shift between both clocks is compensated.

Through the use of multiple DCMs, different clock domains inside the hardware firewall can be generated. The data delivery between these clock domains is realized with clock domain crossings. That means, synchronization FIFOs, which have separate read and write clock input ports, are used. If, e.g., request data is sent to the DDR controller, this data is written with a speed of 150 MHz in the DDR controller request synchronization FIFO. After that, the 200 MHz clock reads the request data from the same FIFO. The answer from the DDR controller is written with 200 MHz into the answer synchronization FIFO and read with 150 MHz. Multiple use of DCMs together with clock domain crossings allow to achieve the maximum speed for each subsystem.

V. CONFIGURATION SOFTWARE

Via a web interface, customers can set their own filtering rules. Before these rules are applied, they are verified by the ISP. The configuration of the hardware is done by platform

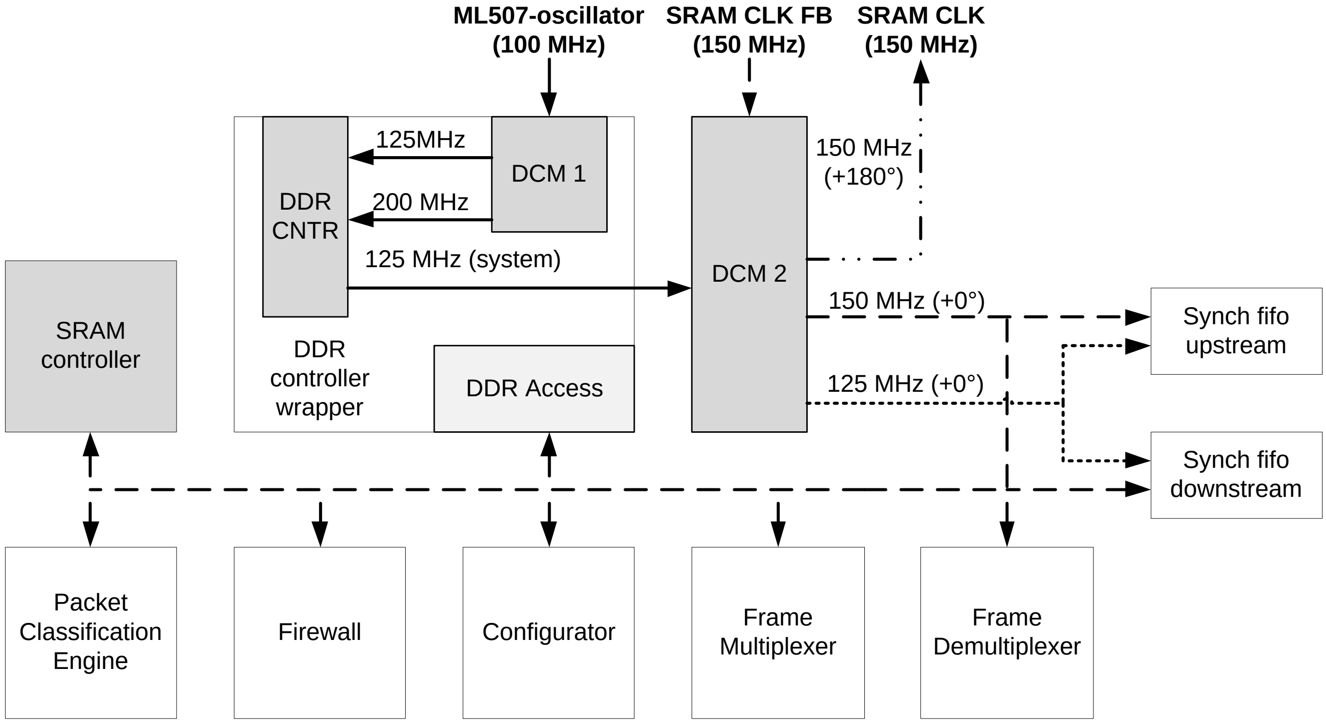

Figure 8. Clocking scheme of the Secure Access Node packet filter

independent software developed with QT. The graphical user interface (GUI) consists of a framework, which is able to include so called plugins. Each plugin offers a GUI to configure a separate hardware component of the Secure Access Node. When starting the GUI, the software searches in a special directory for available plugins. All plugins are loaded and appear in the software as a tab. By means of the plugins, ISP provided rule can be generated and customer rules are applied. Furthermore, the configuration software is able to interrupt the hardware processing flow for updating the hardware configuration.

VI. Conclusion

Because many subscribers do not have the necessary knowledge to maintain their own security measures, it is important to include security features at the ingress of the network. Therefore, we have designed a software/hardware co-design consisting of a packet filter firewall, a signature detection, and a Web filter module. The implementation results show a reachable speed of 150.3 MHz corresponding to 4.81 Gbit/s. Furthermore, subscribers are protected by the Secure Access Node and do not need to care about their own security. Especially for the large number of customers with minor technical knowledge, this is an important feature. Because of the applied methods, the bandwidth of customers is not influenced. Furthermore, no attacker has access to the hardware. Only an ISP administrator is able to update the security mechanism. Moreover, it is possible to update the system during operation. Prospectively, a functional test with real traffic data is intended.

As many subscribers do not have the necessary knowledge to maintain their own security measures, it is important to include security features at the ingress of the network. Therefore, a hardware-based approach consisting of a packet filter firewall, a Web filter module, and a signature detection engine is presented. As a hardware solution, it offers more advantages in terms of security and robustness. The implementation results show a reachable throughput of 4.81 Gbit/s for the packet filter firewall, 3.58 Gbit/s for the Web filter module as well as 1 Gbit/s for the intrusion detection engine. The throughput is only limited by the FPGA type and can be even multiplied by using application-specific integrated circuits.

Furthermore, subscribers are protected by the Secure Access Node and do not need to care about their own security. Especially for the large number of customers with minor technical knowledge, this is an important feature. Because of the applied methods, the bandwidth of customers is not influenced. The configuration of the suggested security system can be done only by the network administrator. Since the Secure Access Node is fully transparent for all network participants, it is safe from attacks. Moreover, it is possible to update the system during operation.

Prospectively, a functional test with real traffic data is intended.

Acknowledgment

We would like to thank the Broadband Access Division of Nokia Siemens Networks in Greifswald, Germany for their inspiration and continued support in this project. This work is partly granted by Nokia Siemens Networks.

References

[1] J. Rohrbeck, V. Altmann, S. Pfeiffer, D. Timmermann, M. Ninnemann, and M. Roennau, "Secure Access Node: an FPGA-based Security Architecture for Access Networks," *The Sixth International Conference on Internet Monitoring and Protection (ICIMP 2011)*, pp. 54–57, 2011.

[2] D. Taylor and J. Turner, "Scalable packet classification using distributed crossproducting of field labels," *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, pp. 269–280, 2004.

[3] A. Guruprasad, P. Pandey, and B. Prashant, "Security features in ethernet switches for access networks, TENCON 2003, Conference on Convergent Technologies for Asia-Pacific Region ," pp. 1211–1214, 2003.

[4] Nokia Siemens Networks GmbH & Co KG, "Multi-Service IP-DSLAM SURPASS hiX 5622/5625/5630/5635 Release 3.8M," January 2012. [Online]. Available: http://www.itm-group.com/web/fileadmin/itm/datenblaetter/NSN/hiX-56xx.pdf

[5] Xilinx, "Platform User Guide rev3.1.2," January 2012. [Online]. Available: http://www.xilinx.com/support/documentation/boards_and_kits/ug347.pdf

[6] G. S. Jedhe, A. Ramamoorthy, and K. Varghese, "A Scalable High Throughput Firewall in FPGA," *16th International Symposium on Field-Programmable Custom Computing Machines*, pp. 43–52, Apr. 2008.

[7] W. Jiang and V. K. Prasanna, "A FPGA-based Parallel Architecture for Scalable High-Speed Packet Classification," *20th IEEE International Conference on Application-specific Systems, Architectures and Processors*, pp. 24–31, Jul. 2009.

[8] M. Dixit, B. V. Barbadekar, and A. B. Barbadekar, "Packet classification algorithms," *IEEE International Symposium on Industrial Electronics*, no. ISIE, pp. 1407–1412, Jul. 2009.

[9] K. Palanisamy and R. Chiu, "High-Performance DDR2 SDRAM Interface in Virtex-5 Devices, rev. 2.2," January 2012. [Online]. Available: http://www.xilinx.com/support/documentation/application_notes/xapp858.pdf

[10] R. Clayton, "Anonymity and traceability in cyberspace," *ACM SIGACT News*, vol. 36, no. 653, pp. 115–148, Nov. 2005.

[11] US District Court for the Eastern District of Pennsylvania, "CDT, ACLU, Plantagenet Inc. v Pappert," *337 F.Supp. 2d 606*, September 2004.

[12] R. Clayton, S. J. Murdoch, and R. N. M. Watson, "Ignoring the great firewall of china," *6th Workshop on Privacy Enhancing Technologies*, no. 16, June 2006.

[13] Verisign, Inc., January 2012. [Online]. Available: www.verisign.com