# Security analysis of Croatia's receipt registration and verification system

Stjepan Groš

University of Zagreb, Faculty of Electrical and Computing Engineering,
Unska 3, 10000 Zagreb, Croatia
E-Mail: stjepan.gros@fer.hr

***Abstract*—Beginning with 2013. a law in Croatia come into the force that requires owners of restaurants, café bars, and similar types of businesses that work with cash to register every receipt with a Tax Administration servers before issuing it to a customer. For the purpose of implementing the law APIS-IT, a Croatian IT company, developed a protocol based on XML, SOAP, and public key cryptography. They also implemented the server side system. It is a well known fact that developing protocols in general, and security protocols in particular, is a very tricky endeavor in which even the security professionals make mistakes. In this paper a security analysis of the protocol for receipt registration, the components of the system, and implementations is presented. Note that this is only a partial analysis, based on publicly available information, which doesn't include testings on live systems due to being illegal by the new Criminal law in Croatia. We identified two weaknesses of the current system. But the main problem of the system is the fact that many business owners are now open to different attacks and nothing has been done to remedy that situation. This is actually a broader problem since, with ever increasing number of on line services nothing is done to increase security awareness of people.**

***Index Terms*—security,xml,threats,finance,analysis**

## I. Introduction

Beginning with 2013. a law [1] in Croatia come into the force that requires owners of restaurants, café bars, and other types of businesses that work directly with cash to register every receipt with the Tax Administration (TA) servers before issuing it to a customer. This is only a first phase of system deployment and during 2013. the rest of the businesses will also follow. For the purpose of implementing the law, APIS-IT, a Croatian IT company, developed a protocol based on XML, SOAP and public key cryptography, along with a server side system. It is a well known fact that developing protocols in general, and security protocols in particular, is a very tricky endeavor in which even the security professionals make mistakes. In this paper a security analysis of the protocol for receipt registration, the components of the system, and implementations is presented.

The paper is structured into 6 sections. First, in Section II we present the motivation for the introduction of the system for receipt registration. We also give an overview of its architecture and protocols used for communication. Finally, we get an overview of controls used to protect the system from deception. Then, in Section III we give a threat model, i.e. the model in which we are going to analyze the security of the system. In Section IV we enumerate different attacks against the system that allow different threat agents to circumvent controls and gain some advantage. This analysis is done under the assumption of the perfect implementation, i.e. no programming or configuration errors in the implementation. But, in Section V we review some problems that might arise in case there are errors and omissions in implementation or deployment. Finally, the paper finishes with the Section VI in which we give conclusions, recommendations for improvements and outline the further work to be done.

## II. The system under the security analysis

The system under the security analysis, i.e. the *Fiscal system*, was developed to prevent fraud that was based on the manipulation with receipts done by fraudulent businesses. In essence, receipts weren't issued, or if issued they were easily erased from the cash registers since they've only been stored locally. There were a control mechanisms in place to prevent such scenario, i.e. the customers could send receipt numbers to the TA which would verify that those receipts are really reported, and also TA could do inspections of both businesses and customers. Namely, customers were required by the law to take receipts. But this control mechanisms were inefficient, i.e. customers didn't report receipts, even if they did TA had problems verifying reported receipts because of diverse ways of enumerating them and manual work involved, and inspections are inefficient and costly. Thus, the Fiscal system was introduced with the following goals:

- Having each receipt stored on TA's servers at the moment receipt is issued, in order to prevent manipulation, i.e. to assure non repudation, integrity and authenticity.
- Unifying receipt numbering schemes in order to make receipt checking easier and possible to automate.

Note that inspections by TA are still necessary in order to enforce receipts being issued. Furthermore, because of resource constraints in terms of a number of available inspectors, the customers are also encouraged to send receipt IDs to TA. This is further incentivized with different prizes that customers can win.

Based on these requirements, the architecture was developed along with appropriate protocols.

In essence, the architecture of the fiscal system is relatively simple. It consists of two entities, a client machine which submits receipts and TA's server which accepts receipts, records them and issues signed response (Figure 1).

The messages are exchanged using SOAP over HTTPS, while the data is serialized in XML format. To ensure authenticity, integrity and non-repudiation of the messages public

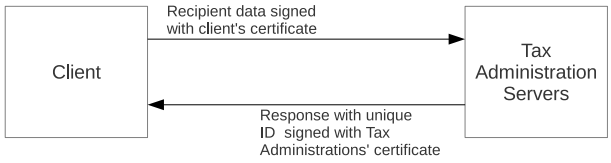


Fig. 1. The architecture of the fiscal system

key cryptography is used. Each entity in this system is issued a certificate by Financial Agency (FINA) [2] which acts as Government's Certificate Authority. The messages are signed before being sent and this signature is generated according to the XML Security specification. To ensure confidentiality (and integrity) of messages while being in transit, SSL/TLS is used. In total, six messages are defined:

- Receipt request/response pair of messages used to register receipt with TA.
- Request/response pair used to register office space.
- Simple echo request/response messages used to test TA's server availability.

Office space registration is used, presumably, for two purposes. First, for TA to know where receipts are issued, and secondly, to restrict the use of a single receipt on different locations.

But, of those, we are interested only in the first pair of messages used to register receipt with TA. So, each generated receipt has to be sent to TA prior to being issued to a customer, to obtain *uniqe receipt ID*. In this paper we will use abbreviation *JIR* for this unique receipt ID as it is so designated in appropriate documentation. An example of a JIR is:

```
6b7749c6-56c1-4cf5-b7f7-9f29cebc9f7f
```

Which is basically standard UUID format. Along with JIR, each receipt also has an *issuer's protection code*, we'll use abbreviation *ZKI* from now on as it is used in the Croatian documents. This code is generated by concatenating OIB, timestamp, receipt ID, office space ID, cash register ID, and total sum, then signing all that and finally taking MD5 of the signature data. Note that ZKI is generated by cash register without any data or involvement from TA' servers. An example of an ZKI is:

```
a1e6b1428f0cc755f0c82aa7a1327e35
```

The motivation for introducing this code is to protect issuer from a malicious third party [3] and also in cases when the issuer doesn't have connectivity to TA, then he can use this code instead of JIR.

There are two controls that force business owner to issue proper receipt:

1) A customer that checks receipt data on TA.
2) Direct controls by TA's inspectors.

Customers are required by law to take receipt with them. Additionally, they can also check JIRs by submitting them on TA's Web pages (or via SMS messages). In that way business owners are forced to issue receipt and they are prevented from falsifying JIRs. Additionally, inspectors serve as control of both business owners (which are required by the law to issue receipt) and customers (which are also required by the law to take receipt). But, the reality is that there are not a lot of inspectors, their knowledge and equipment is rather poor. Customers, on the other hand, don't usually take receipt. This makes this control mechanisms rather weak.

Customers can verify receipt using either SMS message or a Web application [4] developed specially for that purpose. In both cases, the customer sends the following information:

- Either JIR or ZKI.
- Date and time when receipt was issued.
- Total amount.

If the entered data is correct, i.e. TA has data about the receipt, the Web application will only give information that receipt is correct. If the receipt can not be found in the TA's database the system will give to a customer a message to try again later. Note that due to the length of JIR and ZKI this control is weakened since retyping those codes is very hard, especially with SMS messages. Thus, errors are very likely and this further incentivizes customers *not to* verify receipts.

Note one interesting question. ZKI looks harder to read, but is shorter. JIR, on the other hand, is easier to read, but longer to type. So, it is reasonably to believe that customers will type ZKI more frequently, even though they are the same in term of the numbers and letters and the only difference is in dashes.

## III. Threat model

As a first step we have to establish a threat model, i.e. a model that defines what are the threat sources and associated threats that can impact a system. Threat sources we analyze are:

- Business owners trying to circumvent paying taxes by manipulating receipts as they used to do.
- Individuals and crime organizations trying to take financial advantage of the system.
- Political groups and movements, like Anonymous, trying to subvert a system for PR purposes.

*Business owners* have incentive to manipulate system in order to avoid paying taxes. The ultimate goal is not to register smaller sum of money than there actually was. This can be done in two ways, (i) by not issuing receipt, or (ii) by somehow manipulating receipts themselves. Since not issuing receipt has nothing to do with the security analysis we'll ignore that scenario so we are only interested how business owner can manipulate receipts. We also ignore non-technical manipulations, like manipulations with fake receipts (quotes). We also assume that business owners have a complete control of cash register, both hardware, network connectivity, and software.

One thing to note is that business owners can be different in size, what will have impact on their network topology and controls they implement. Here, we'll assume small business owners, i.e. owners of restaurants, café bars and similar types of businesses.

It is the fact that cash register now has to be connected to the Internet (it has to be in order to be able to communicate with

Tax Administration's servers). Some of those will be directly accessible on the Internet (by having public IP address). Many will be run by Windows operating system, and thus will be threatened by usual threats, like mallware. Last, but not least, probably business owners will offer their customers to access the Internet over the same LAN they use for communication with Tax Administration. This connectivity increases number of potential attackers, which we grouped into two categories, according to their motives for the attacks.

The first category consists of *individuals and crime organizations*. The difference is in the resources available to mount an attack. Obviously, individuals have fewer resources, while crime organizations have much larger resources on their disposal. So, individuals can try to attack at most few cash registers, while crime organizations can attack as much as they want.

The second group are political groups that aren't so much interested in financial gain but are satisfied with PR like events, like DoS attacks.

In the majority of the following text we'll assume *perfect implementation*, that is, will assume that programmers, both on server and client side, that implemented the protocol didn't make any programming mistakes. This is obviously very unrealistic assumption, especially from the point of business owners. At the end we'll outline possible mistakes that programmes could make and what this means in terms of attack possibilities. Still, this is a topic for a separate research.

## IV. Attacks

### A. Attacks by business owner

As we saw in Section II there are two primary controls that check invoices issued by business owners, customers and inspectors. Customers can not access cash register in order to check if invoice is handled properly or not, while TA's inspectors can to some degree.

So, here are some possibilities what business owners can do in order *not to* register valid receipt with the TA while in the same time avoiding being caught by the customers reporting invoice data to TA:

- Business owner can generate receipt only with ZKI. Then, he observers if the customer takes receipt or not. If he takes it then the Business owner registers receipt with TA, otherwise, he removes receipt from the system.
  Note also that ZKI can be a random number, there is no way for customer, or TA, to check if it is valid or not. That means that if TA receives receipt with only ZKI it *can not prove* that this receipt was issued by certain business owner. This relates to the verification of receipt via Web or SMS. In that case many businesses can coordinate to issue receipts with the same ZKI, amount, date and time. Alternatively, they can reuse one's receipt.
  Note that it is relatively easy to force customers to use ZKI instead of JIR. All they have to do is to print some random JIR. If the customer types JIR, the error will be reported and finally, if the customer is persistent enough he will finally try with ZKI.
- The way receipt checking with TA is implemented opens up possibility of reusing JIRs. Business owner can issue one receipt that will be registered with TA's servers. Then JIR and ZKI can be reused for other receipts with the same amount within a certain time frame.
- Since either JIR or ZKI is typed when the receipt is verified, this means that they don't have to be correlated. In other words, you can write JIR and ZKI belonging to a different receipts. But, for the time being it is questionable if this opens up some new possibilities to avoid registering invoice with TA.

Note that if issuing multiple receipts with the same JIR/ZKI then care must be taken about invoice numbers which are required to be sequential.

Now, second control, i.e. TA's inspectors, complicates things a bit, but not much. The first avoidance tactic in the previous list is the most dangerous. Namely, it rests on the fact that two, or more, persons don't collude, that is, they don't compare their invoices. TA's inspectors might do that, that is, they might collect somehow few invoices and if any two have the same JIR or ZKI, obviously the business owner has tried to circumvent the system.

The key difference between customers and TA's inspectors is that the letter can inspect data on cash register to some degree. What could happen is:

- Inspector comes with a bunch of invoices and asks business owner to show them records in cash register. If there are multiple invoices with the same JIR/ZKI/number then the business owner must be very careful which one he will show to inspectors. But, in the end, inspectors have no way to detect fraud *unless* they have at least two invoices with the same JIR/ZKI/number.
- For receipts that reuse JIR/ZKI numbers inspectors might notice that there is some discrepancy between dates on receipts and the actual time they were issued. But, there is no requirement on time synchronization on cash registers.
- The specification of the fiscal system defines that tax administration might ask business owner to recreate (or restore) ZKI. But it is very hard to verify that ZKI was generated using right data and, more importantly, the right certificate.

To conclude, without TA's inspections business owners have opportunities for manipulations even though they are not as easy as they were before. To which extent this manipulations can be done is a topic of a further research, especially if business owners collude. Additional problem is that TA's inspectors have to be properly equipped in order to be able to detect fraud. Finally, some fraud can be detected via advanced techniques like data mining on TA's servers.

### B. Attacks by malicious individuals

Malicious individuals have enough resources to attack one, or at most several cash registers (CRs). Looking again on the Figure 1 individuals are a significant threat to business owners, and a lot less to TA's servers. Some of their motives can be the following ones:

- To harm competition by manipulating competition's receipts and in that way incur penalties by TA.
- To reuse invoices by competition and in that way no to register their invoices.
- To monitor what competition has been doing.
- Individuals that just want to do it because they can, to show off themselves.

In order to attack business owner and their CRs individuals have to be either:

- Somewhere on the communication channel, i.e. trying to perform MITM type of attacks.
- Trying to compromise cash register machine itself with a final goal of obtaining the secret key belonging to the business owner.
- Denial of service attack on Internet connection and/or cash register machine.

The communication channel is protected using HTTPS, so in case there are no bugs in the implementations it is very hard for the attacker to break into that channel. Note that two controls are of utmost importance here. The first one is that communication is performed via HTTPS (i.e. no fallback to HTTP) and second that the certificate check is properly implemented. The only way attacker can break into the channel is by exploiting bugs in the implementation. Since we assumed that there are no bugs, than this isn't an issue. Still, we'll return back to this possibility in Section V.

The second attack vector is by exploiting the client machine, i.e. cash register, itself. The grand prize in this case is stealing private key. The moment the attacker has control of the machine he is basically in a possession of private key. There could be protection in form of a password on a file but this password has to be stored somewhere. The ways in which an attacker can achieve that are usual attack vectors, of which the most dangerous one is social engineering attack. What makes things even worse here is that cash registers were old machines with old hardware and not maintained well. Now, when those machines are on the Internet, with employees that will probably use them to surf the Web which opens up a lot of possibilities to attack the machine itself. This is, in a way, *game over* situation in which attacker can do whatever he/she likes.

The third attack can prevent business owner from operating and can degrade him into the mode where it only issues ZKI, not JIR. The additional problem with this type of attack is that it manifests itself indirectly (e.g. error message failed to establish connection) and business owner can not know that he is actually under the DoS attack.

To conclude, malicious individuals are a very serious threat and to protect themselves from this attacks business owners have to have properly maintained machines as well trained stuff. Unfortunately, in a situation in which business owners, for whatever reason, try to save as much as possible it is highly likely they will not invest in security. Additionaly, they don't have a habit of maintaining machines. What TA has to do, it has to educate business owners and warn them about threats they face.

### C. Crime organizations

Crime organizations have enough resources to attack potentially many business owners and their primary goal is a financial gain. Basically, what they can do is attack many business owners simultaneously and for that purpose they have same attack vectors at their disposal as individual attackers. But, additionally, they have one more attack vector and that is malicious software specifically tailored for this purpose. The problem with such approach, i.e. tailored malicious software, is that standard antivirus software is lot less efficient and this poses a significant problem.

Additionally, it is possible to attack TA's servers via some innocent clients and in that way to hide real attacker. This is viable threat under the assumption that there is some implementation error in TA's server, as we further discuss in the Section V.

### D. Activist groups

Activist groups, like Anonymous, are attacking for the primary purpose of publicity, not financial gains. Thus, attacking a single business owner doesn't give them sufficient pay offs. It is more likely they will try to attack TAs servers with the primary goal of causing disruptions to the service.

This is actually very real threat because it is relatively cheap to by services from some bot owner and do prolonged DDoS attacks on the TA's servers themselves.

## V. Implementation based problems

In the previous analyses we assumed that the implementations are perfect, both on the client side as well as on the server side, i.e. TA's servers. Unfortunately, it is very likely that there will be many errors, especially on the client side as there are a number of different implementations done by people that don't usually do security related protocols. In this section we are going to list some of the potential problems that might arise from different mistakes done by implementors, but in general this is open to further research.

The first and foremost question is whether clients require https connection to server and do they correctly check server's certificate. If either of those isn't fulfilled, then MITM attack is possible. At minimum, an attacker that successfully intercepts communication channel can see messages between client and server and in that way confidentiality of the business owner is violated. The next thing the attacker can achieve is to disrupt communication, either by not forwarding messages or by modifying them. Note that we assume that modifications will be detected by server or client, depending who's receiving them.

There is also question about covert channels which are very dangerous as they allow information leak. In the extreme, it would allow someone to deduce TA's secret key which would be the ultimate attack on this whole system. Note that attacks on TA's servers are supposedly protected by having each business owner sign an agreement with TA/FINA. But, this doesn't preclude some attacker from compromising business owner's machine and performing attacks using it as a proxy.

Many other potential problems here exists, like proper XML processing, schema validation, etc. To conclude, there are many potential dangers here and this is a topic for further research.

## VI. Conclusions and future work

In this paper we presented initial analysis of a fiscal system introduced by Republic of Croatia's Tax Administration. It is only a partial analysis, based on publicly available information, which doesn't include testings on live systems due to being illegal by the new Criminal law in Croatia. Also, we assumed that the implementations are perfect, i.e. there are no software bugs.

The finding is that on the server side the system is relatively well protected. On the client side, the things are quite different. This is Achilles' heel of the whole system. We think that by introducing this whole fiscal system, without warning and proper education of business owners many of them are brought into a dangerous situation. It is the truth that it is the consequence of irresponsible behavior of individuals, which don't care about security, but it is also a Government's irresponsible behavior that neglects the other side. So, some actions have to be taken here.

There is also a weaknesses in the control part of the system are, the one that forces business owners to issue receipts, and as usual, the weakness is on the humans. Namely, it is questionable how much customers will be willing to type JIR or ZKI to check the receipt. Furthermore, it is mandatory that TA's inspectors be properly educated and equipped.

During the threat analysis we identified two technical weaknesses. First, there is possibility to reuse JIR's by manipulation the date and sum on receipts. That one is very hard to prevent without rigorous controls, which are infeasible. The second weakness is ZKI, which by itself is useless. There is no way for TA's inspectors to *prove* it has a correct without having private key, which by definition they are not allowed to have!

Fortunately, this systems uses a lot of advanced technology, primarily cryptography, which many people don't know much about, and this for the time being acts as a barrier for more frauds.

As for the future work, we think that the main research should be done on live systems to identify weaknesses they have. More specifically, it is our intention to build a proxy that would automatically test certain implementation for correctness. Next, a motive for attackers is not elaborate enough, but this requires financial along with technical knowledge. Finally, we think that the introduction of advanced statistical analyses on TAs servers might help in detecting some manipulations described here, i.e. in order to spot strange behavior in the registration process. But, this is a topic for a further research.

## References

[1] "Odluka o proglašenju Zakona o fiskalizaciji u prometu gotovinom (in Croatian)," Nov. 2012. [Online]. Available: http://narodne-novine.nn.hr/clanci/sluzbeni/2012_12_133_2822.html

[2] "Financial agency," last checked: February 6th, 2013. [Online]. Available: http://www.fina.hr/Default.aspx?sec=1134

[3] "Fiskalizacija - Tehnička specifikacija za korisnike - Verzija 1.2 (in Croatian)," Dec. 2012. [Online]. Available: http://www.porezna-uprava.hr/fiskalizacija/dokumenti/Fiskalizacija%20-%20Tehnicka%20specifikacija%20za%20korisnike_v1.2.pdf

[4] "Web application for invoice checking," last checked: February 6th, 2013. [Online]. Available: http://www.provjeri-racun.hr/provjeraracuna