E-Receipt Verification System with Elliptic Curve Digital Signing Algorithm for Receipt Forgery Prevention

Muhammad Harits Shalahuddin Adil Haqqi Elfahmi (*Author*)
Informatics/Computer Science, School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
adilelfahmi@gmail.com

***Abstract*—This paper proposes a new e-receipt system that will use digital signature to verify the authenticity, integrity, and non-repudiation aspect of a receipt. This system will utilize a third party that offer a service that receives a customer-approved receipt from the merchant and forwards the digitally signed receipt to the user handheld device. The system is designed such that it can prevents receipt forgery to perform redemption fraud and return fraud.**

Keywords—E-receipt; System; Digital Signature; ECC; ECDSA; Forgery; Prevention; Handheld device; Redemption Fraud; Return Fraud;

I. INTRODUCTION

Receipt has long been used to record the proof of transaction. In developing countries where minimart is rising greatly in number, every buyer at minimart would automatically handed out a receipt as their proof of their transaction, no matter how small the transaction value is. In reality, the buyer doesn't always need the receipt. This makes the receipt paper handed to the buyer will goes to trash for most of the time. Such little trash will becomes bigger when taking account hundred thousand numbers of minimart available across the country.

On the other hand, while receipt has been used for a long time, it still has it flaws. Simple receipt would have been easy to change or fake to benefit the perpetrator needs. For example, a company that assign its worker to buy something expensive could be fooled when the worker gave them receipt with fake higher listed expense, where the difference in cost will goes to the worker's wallet.

To solve the problems mentioned above, this paper will try to create a system that will utilize a third party signing service provider that will replace conventional paper-based receipt into a digitally signed receipt. The digitally-signed receipt will be saved on the client application on the user handheld device for easier management. The application will also have verification feature to prevent any forgery or fraud.

II. LITERATURES STUDY

A. Receipt

Receipt is defined as a "a writing acknowledging the receiving of goods or money" or "a piece of paper on which the things that you buy or the services that you pay for are listed with the total amount paid and the prices for each" [1]. Example of a paper-based receipt can be seen at Figure II.1.

George's Restaurant
300 72th Street
Miami Beach fl 33141
305-864-5586

Server: Ronald	01/19/2013
Table 20/1	10:53 PM
Guests: 1	10062
Reprint #: 1	
Ferrari Carano	47.00
Insalate Cesare	7.50
Caprese with prosciutto	9.50
FISH SPEC	25.95
Spinach Ricotta Ravioli	15.95
Seafood Pasta	19.95
Ossobucco	29.95
Sub Total	155.80
Tax	14.02
Total	169.82
Balance Due	169.82

GRATUITY NOT INCLUDED!!!
Thank you for your business

Figure II.1 Example of a paper-based receipt (source: tripadvisor.com)

As with other methods related to payment, there are various way a receipt can be used to do fraud. There are two forms of mostly used fraud involving receipt. The first one is the examples of a return fraud. Return fraud is the act of

defrauding a retail store via the return process. One of return fraud example is receipt fraud. It is an act of utilizing reused, stolen or falsified (forged) receipts to return merchandise for profit. Alternatively, returning goods purchased on sale or from a different store at a lower price with the intention of profiting from the difference. Another example of return fraud involving receipt is e-receipt fraud, which is basically the same with receipt fraud but using e-receipt. Return fraud has cost seller as much as 9.1 billion dollars in 2013. [2]

The second form is when a company asks its employee to buy some product and will redeem the money used to buy the said product. The fraud is done when the employee tries to fake/forge the receipt received from the product seller and make duplicates with higher listed cost. This way, the employee will have more money to redeem at their company, and the company won't be able to tell the difference. This method can be done for both paper-based receipt and e-receipt.

B. Elliptic Curve Digital Signature Algorithm (ECDSA)

Digital signature is one of information security technique used to provide authenticity, non-repudiation, and integrity for a digital content. For example, a sender Alice sends a message to Bob. As receiver, Bob can prove that the message truly sent by Alice and Alice can't deny that she sent that message to Bob. Digital signature also can guarantee that the message which Bob received is exactly the same that Alice sent

There are some digital signature algorithms, one of them is *Elliptic Curve Digital Signature Algorithm* (ECDSA). ECDSA is a lightweight signature algorithm because it operates in elliptic curve group. ECDSA uses multiplication in elliptic curve, which is same as repeated additions of two points, as a basic operation. In general, there are three main components in every digital signature algorithm those are *key generation*, *signature generation*, and *signature verification*.

Key generation is the first thing to do when using digital signature. When a sender Alice sends a message to Bob, they must agree on a set of domain parameters of the elliptic curve that they will use later. As a sender, Alice must have a private key *dA* (a random value less than *n*, *n* is the order of the curve). Alice must keep the private key to herself. Alice also have a public key *QA* (the value of *QA* depends on the private key, QA = dA * G, G is a generator point). Alice can share her public key to the receiver for verifying.

When Alice sends a message, she will sign it with a function called signature generation. There are five steps in generating an ECDSA signature:

1. Calculate hash value of the message *m* with one hash algorithm, for example SHA1 (as used in this implementation) $e = HASH\ (m)$.
2. Find a random integer k in $[1,n-1]$
3. Calculate $r = x1\ (mod\ n)$, where $(x1, y1) = k * G$
   If $r = 0$ back to step 2.
4. Calculate $s = k - 1(e + dAr)(mod\ n)$.
   If $s = 0$ back to step 2.
5. The signature of a sender Alice is a pair $<r,s>$

When receiving a message, Bob can prove that the message comes from Alice and it was not altered during the sending process with signature verification. There are six steps:

1. Verify that the value of *r* and *s* are in $[1,n - 1]$. If not, the signature is invalid.
2. Calculate $e = HASH\ (m)$, with the same hash algorithm in signature generation
3. Calculate $w = s - 1\ (mod\ n)$
4. Calculate $u1 = ew\ (mod\ n)$ and $u2 = rw\ (mod\ n)$
5. Calculate $(x1, y1) = u1G + u2QA$
6. Signature is valid if x1 = r(mod n), otherwise the signature is invalid.

III. Proposed Analysis And Design Solution

As mentioned in section II.A, several fraud methods can be used to a receipt to gain benefit by the fraudster. We will start by pointing out vulnerable parts present in the current paper-based receipt design system and analyze it. After that, we will come with appropriate countermeasures and propose a new receipt system based on the countermeasures.

A. Conventional Paper-based Receipt Design System

First, we will analyze the system design involving the examples of return fraud, which is receipt fraud and e-receipt fraud. Both of them involves the buyer and the seller in the system. The design itself can be seen at Figure III.1.

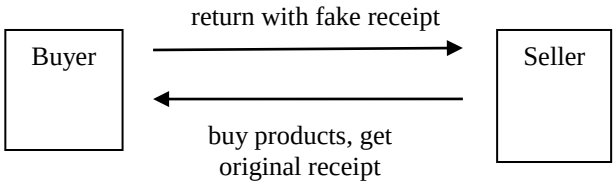


Figure III.1 The system design involving receipt and e-receipt fraud

The problem with such design is when the seller cannot verify the authenticity of each product price, it is vulnerable to fraud. While recently seller has been using database to manage their products and having easier time to detect inconsistencies in price value, it is still rather inefficient to check every product price one by one, instead of checking the authenticity the receipt itself.

The second form of fraud to analyze is the design that involve three party, which is the seller, the buyer/employee, and the company. The company is the party that asks the employee to buy products on the seller. The employee buys the product from the seller and receives the receipt. The employee forge the receipt and uses it to redeem more money than used to the company. We shall call this fraud the redemption fraud throughout this paper. The design can be seen at Figure III.2.

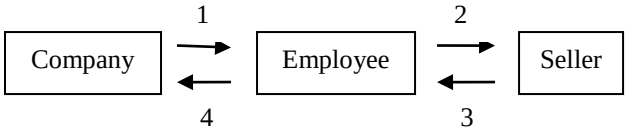


Figure III.2 The design system involving the redemption fraud

corresponding meaning:

1) The company asks the employee to buy some products
2) The employee buys the products on the seller
3) The employee gets the original receipt. After receiving the receipt, the employee will further forge the receipt so that it will have higher price value
4) The employee gives the product and redeem the forged receipt to the company. The employee profits from the difference between the original receipt and the forged receipt.

The problem with this design is that sometimes, the employee can't be perfectly trusted. In process (3), they can forge and fake the receipt so it can list higher cost. Because of that, we need to make sure that the receipt that comes from the seller still remains the same even when it comes to the redemption process at the company.

B. Proposed E-Receipt Design System

From the last section, we can find that we need a mechanism for the seller/company to check that the receipt remains the same even after the buyer already receive it. While the seller can check their database to check every item individual price, the company will have trouble doing so. In that case, this design system will focus more on the countermeasures for the redemption fraud scheme. Although it focus on it, this system will also be applicable for the return fraud scheme.

E-receipt is a strong tool that can be used to solve this problem. One of the advantages of e-receipt is that it uses less paper and easier to carry (inside handheld device). Although it suffers the same problem as paper-based receipt, we can cover that weakness with digital signature.

When using e-receipt there are advantages that can be achieved more that authenticity and integrity check, which is the less use of paper. For seller that automatically hands out paper receipt to the buyer, it can cut costs in buying the paper material for the receipt itself. Furthermore, for convenience, the e-receipt that given out by the seller can also be inserted to the buyer handheld device. This way the buyer can easily manage their e-receipt anywhere anytime without have to bring their paper receipt everywhere.

There are many kinds of digital signature methods. One of them is Ellipse Curve Digital Signature Algorithm. This method has the advantages that it uses shorter key but have the same performance as normal digital signature algorithm. Using ECDSA, we can easily sign and verify an e-receipt without worrying whether it would takes up too much memory and performance in the buyer handheld device. This makes ECDSA suitable for signing e-receipt into user handheld device.

Using e-receipt and ECDSA, we propose the following e-receipt verification system. There are two additional things that will be added to this system, compared to the conventional system. First, a trusted third party service that can: (1) receive a plaintext receipt data from the seller device, (2) send digitally signed e-receipt to the user handheld device, and (3) act as the verifier whether an e-receipt is valid. Second, a client application on the user device that can: (1) receive a signed e-receipt from the service, and (2) asks the service whether an e-receipt is valid. The interaction between actors on this system can be seen on Figure III.3.

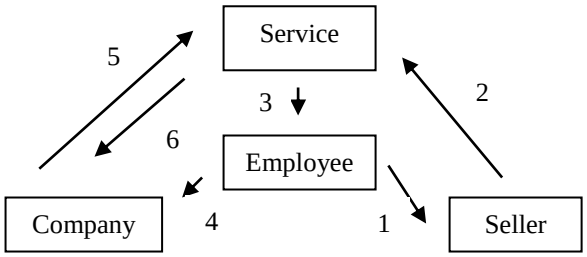


Figure III.3 The design system of the proposed e-receipt verification

Before the process starts, there are prerequisites. The employee (the buyer) and the seller must have already registered in the service, and after that the process can finally start. From Figure III.3, each process number have the corresponding meaning:

1) The employee gives the ID to the seller through the user device. The client application on the user handheld device will have this feature.
2) The seller will send the triplet (Buyer ID, Plaintext receipt data, Seller private key) to the service. The buyer ID retrieved at process 1, while the seller private key will be saved on the seller order management application.
3) The service will process the plaintext data and sign it with private key from the seller. After that, it will push the signed e-receipt to the buyer using the buyer ID received from the seller.
4) After the employee receive the signed e-receipt, they can give it to the company using the same application client. Sending e-receipt to another user is another feature of this application.
5) At this process, the company begin to verify the integrity of the e-receipt received. To get the verification result, first the company must send the seller ID to the service. By doing this, the client can retrieve any seller public key without knowing them beforehand.
6) The service returns the corresponding public key that matches with the seller ID saved on the service

database. At the seller registration process, the service will generate the public key and the private key for the seller. The private key will be embedded on the seller order management device, and the public key will be saved on the service database. After that, the client application can check the integrity of the signed e-receipt with the public key received from the service.

Using the process above, the company can check the integrity and can confirm that the e-receipt remains the same while being transferred through the employee device. At the same time, to prevent the return fraud mentioned before, the seller can also use the same functionality to check for any changes in the e-receipt given by the employee/buyer. With that reasoning, this system can both prevents either the redemption fraud or the return fraud.

IV. Implementation and Experiments

The client application, the service, and the algorithm will be implemented in JAVA. The client application itself will specifically work on Android. The ECDSA implemented will use as a standard curve which is NIST P-192 [3]. Choosing standard curve makes the client application more secure. Overall, the system works like explained in section III.B. The ECDSA implementation used on the application can be seen at Figure IV.1.

```
function generateSignature(dA, receipt_content, ecc):
   receipt_content ← hash(receipt_content)
   do
      k ← random integer in [1, ecc.getN()-1]
      r ← ecc.times(k, ecc.getG()).x mod ecc.getN()
      s ← k^-1(receipt_content + dA.r) mod ecc.getN()
   while r = 0 or s = 0
   return <r, s>

function verifySignature(public_key, receipt_content, ecc)
   if r and s are not in [1, ecc.getN()-1]
      return false
   else
      receipt_content ← hash(receipt_content)
      w ← s^-1 mod ecc.getN()
      u1 ← z.w mod ecc.getN()
      u2 ← r.w mod ecc.getN()
      (x1, y1) = u1 x G + u2 x QA
      if x1 = r mod ecc.getN() then
         return valid
      else
         return invalid
```

Figure IV.1 ECDSA implementation used on the application

The private key that is generated from the ECDSA will contains the private key itself. The public key that is generated will contain the point of public key itself. The format for the private key and public key can be seen at Figure IV.2 and Figure IV.3.

```
<private key>
```

Figure IV.2 The application private key file format

```
<public key point x> <public key point y>
```

Figure IV.3 The application public key file format

Both the redemption fraud and the return fraud (e-receipt fraud) attack relies on the ability of the buyer to change the e-receipt (forgery) to fit their needs. That is why to test its ability, we only need to check the client application verification process when some aspect of the e-receipt (content/signature) changes. We also need to check the performance of ECDSA on a handheld device compared to normal digital signature algorithm. For more detail, we will perform the following experiments:

1) Change the e-receipt content and verify it
2) Change the e-receipt signature and verify it
3) Change the public key and verify it
4) We will average the verification performance difference for all tests between normal digital signature algorithm compared to ECDSA on the client handheld device. We will use JAVA built-in digital signature algorithm for comparison (1024-bit). [4]

Because we only need to test the fraud aspect of system, we can skip process 1, 2 and 3 on the Figure III.3, and begin the experiment when the receipt is in the most vulnerable state to forgery (after process 3, at the buyer's hand). Because of it, we can use the assumption that process 1, 2 and 3 went fine.

We will also use two set of private and public key for the signing and verification experiments. For experiment (3) we will sign the e-receipt with private key from the first set, and then try to verify it with public key from the second set. The two set of private and public key can be seen at Figure IV.4 and Figure IV.5. The client application interface on the user handheld device used to test the verification process can be seen at Figure IV.6.

```
Private key:
27682240100821392214509349251976158920406693
57440981208673
Public key:
39501281202664861244295806522001340171647587
48531063049804
57061633924567949169185930867839093452035013
67484569588913
```

Figure IV.4 First set of private and public key used for the experiments

Private key:
693796722020073876036298774023672798752418893273754058766
Public key:
429218389207589029995811430226310376444657329731516138780958154691158229386854906989922743049773797174193160540948 84

Figure IV.5 Second set of private and public key used for the experiments

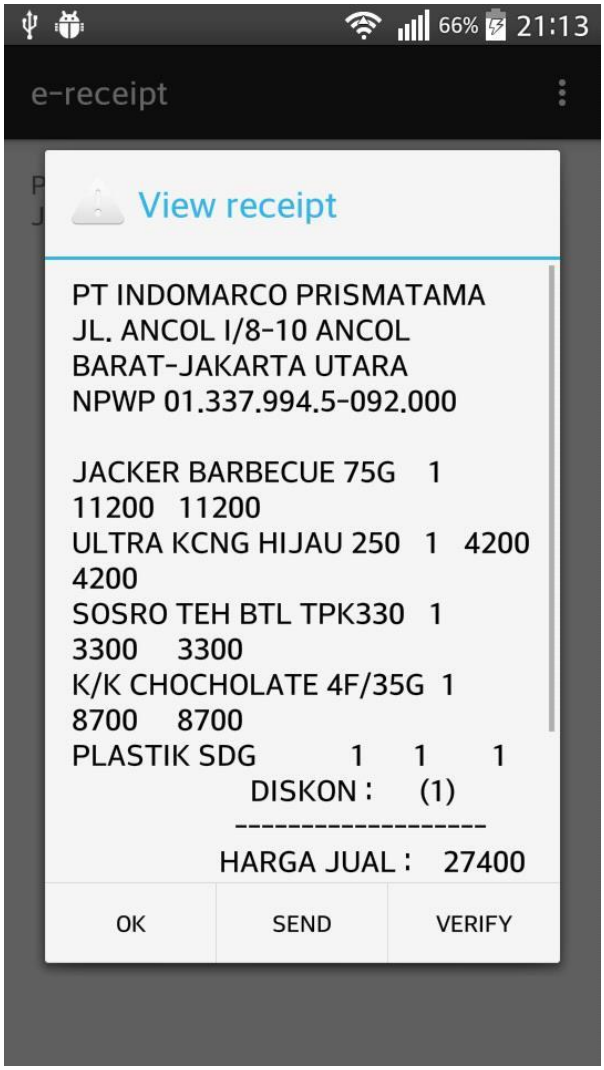


Figure IV.6 The client application interface for verification implemented in Android

V. Experiment Results and Analysis

A. *Experiment Results*

For the tests we mentioned in section IV, we put the result and verification performance to a table. The table containing the results could be seen at Table V.1. Each test number indicates the test explained on section IV. Each label on the table header have the following meaning:

A. Does the e-receipt content changed?

B. Does the e-receipt signature changed?

C. Does the e-receipt public key changed?

D. Is the e-receipt valid (verified)?

E. Performance with ECC

F. Performance with Java DSA

Table V.1 Experiment result table

Test No.	A	B	C	D	E	F
1.	No	No	No	Yes	473ms	1ms
2.	Yes	No	No	No	402ms	1ms
3.	No	Yes	No	No	371ms	1ms
4.	No	No	Yes	No	506ms	1ms
				Avg	438ms	1ms

From the result shown on Tabel V.1, we can see that the digital signature can correctly verify when some aspect of the e-receipt changed. The performance result shows that Java built-in DSA is significantly faster on the Android device. The meaning of these results will further be analyzed on the next section.

B. *Analysis*

From the previous section, there mainly two results that can be analyzed, which is the algorithm reliability and performance. The algorithm proven to be reliable while the performance differs greatly between our implementation of ECDSA and Java built-in DSA.

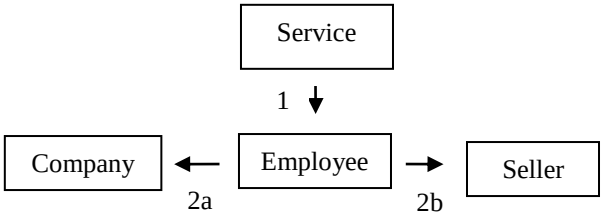


Figure V.1 Possible fraud scheme on the e-receipt

After we confirm the reliability of the verification method, we can safely say that this application can prevent redemption fraud and return fraud mentioned in section III. Both of those fraud scheme can be seen at Figure V.1. At process 1, the service forwards the signed e-receipt to the employee. The

employee then have two choices: (1) do the redemption fraud to the company (process 2a), or (2) do the return fraud to the seller (process 2b). Although the employee have two fraud choices to do, there are one thing in common if they want to do either choice: they must be able to forge and fake the e-receipt. Because we have proven from the experiment result in section V.A that it is impossible changing the aspect of the e-receipt without it becoming invalid, it is impossible to do either of the fraud mentioned before.

As we can also see on the experiment result, the performance of the ECDSA is very slow compared to the Java DSA. Even though the ECDSA implementation uses 192-bit key length and the Java DSA uses 1024-bit key length, theoretically the ECDSA should be faster than Java DSA. The reason of this result is because the ECC class used by the ECDSA that handled all the operation regarding the point uses Java BigInteger class, which is slower than normal mathematical operation normally used in Java. We represent every number and points in our implementation in BigInteger variables. On the other hand, the Java DSA uses byte representation, which is more lightweight but harder to implement and debug when there are errors. With more improvements and better representations we believe that the ECDSA should be faster than Java DSA.

VI. Conclusion

We have shown that there are several fraud method that can be applied on paper-based receipt system. Those fraud method are redemption fraud and return fraud. Both of them involves the buyer (employee) as the fraudster, while the company and the seller being the victim, respectively.

After several experiments, our system has proven to be effective to prevent e-receipt forgery. Because of both of the redemption fraud and return fraud needs to forge the e-receipt, preventing it with our system makes the fraudster unable to do either of those fraud.

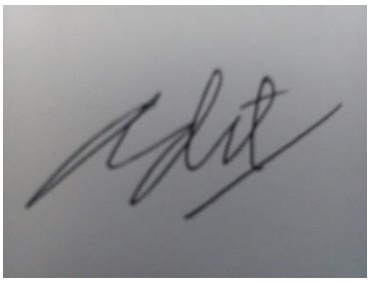
Choosing ECDSA as the algorithm for this system was correct. Because of the limitation of the device, we need to use an algorithm that uses as smaller key as possible without losing security aspect. While our implementation performance loses to Java DSA, we believe that it is only representation issues caused by Java BigInteger class.

Besides of preventing the fraud, there are other advantages this system can offer. The buyer can easily manages their e-receipt without the need of carrying them everywhere like paper-based receipt. The seller also benefit from the less need to spend money for buying paper used to print the paper-based receipt, and thus saving more money.

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 10 Mei 2015



Muhammad Harits Shalahuddin Adil Haqqi Elfahmi 13511046

References

[1] http://www.merriam-webster.com/dictionary/receipt, accessed on 9/5/2015 2:00pm

[2] https://nrf.com/news/return-fraud-cost-91-billion-2013, accessed on 10/5/2015 5:00pm.

[3] http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf, accessed on 9/5/2015, 3:00pm

[4] https://docs.oracle.com/javase/tutorial/security/apisign/gensig.html, accessed on 10/5/2015, 5:00pm