International Journal of Security and Its Applications
Vol.8, No.6 (2014), pp.245-260
http://dx.doi.org/10.14257/ijsia.2014.8.6.22

A Novel Approach towards Detection of Spoofed Manufacturer Identity under the Collaborative Frameworks of Mobile Computing and Product Specific QR Codes

Soham Sengupta[1] and Dr. ParthaPratim Sarkar[2]

[1]*Department of Information Technology, JIS College of Engineering, India*
& Research Scholar, DETS, University of Kalyani, India
soham.sengupta.java@gmail.com; sohamsengupta@yahoo.com
[2]*Professorand Senior Scientific Officer, DETS, University of Kalyani*
parthabe91@yahoo.co.in

Abstract

Today's advanced machineries have immensely contributed to an open, worldwide market for fake products which include not only electronic goods and other everyday commodities, but adulteration has pawed into highly sensitive industries which manufacture medicines, baby-food, beverages; and adulteration in medicines and beverages often turns out to be fatal. The prime force behind the growth of these fraud markets is some highly equipped packaging and printing machinery, which make it too difficult to distinguish between a genuine and an adulterated product by looking at its package. While incidents of currency forgery is very common nowadays, especially in the developing countries, the frequency of tracing adulteration in a bottle of beverage or an ampule of a life-saving drug has got a rapid, concerning momentum. The thesis aims towards detection of a fake product by one's smart phone under a collaborative framework of mobile computing, cryptology and product specific QR codes to be used as the label on the products' package.

Keywords: *Fake product detection, Mobile assisted Verification, QR codes; Android, Zxing*

1. Introduction

Advanced packaging and labelling machineries have been the prime and collaborative factors behind the concerning growth of the market of fake products. Often an adulterated medicine or a bottle of beverage or a branded baby-food packaged with the help of an efficient and clever technology, cannot be distinguished to be not original by simply looking at the packing or the sealed container. The consequence of using a fake medicine or consumption of adulterated food or beverage can often have dire consequence. Considering it to be granted that forgery in packaging and adulteration cannot be feasibly stopped overnight, the aim of this thesis is to present a fast, plausible and cost-effective model that enables a customer to verify whether a product (*e.g.*, medicine, food, beverage or any other electronic and non-electronic product) is original before she makes the purchase. A product packaged with a spoofed manufacturer's identity can thus be detected. The backbone of the model envisaged in this thesis is a collaborative framework of mobile computing, cryptology and QR codes.

ISSN: 1738-9976 IJSIA

2. Proposed model

Our model, at the user's end, is a smartphone application (*e.g.,* an *iOS* or an *Android* application) to verify whether a product is genuine or not. The original manufacturer is required to distribute the mobile application over some secure, trusted application store (*e.g.,* Apple store, Google Play *etc.,*) for free. The manufacture will use a QR code to label the products while packaging. There will be certain information about the product which will be so encoded that it can be read and decoded by the manufacture's application only. In order to ensure best reliability and least vulnerability of this model, the thesis adds the concept of Asymmetric Cryptography. Before generating the QR–labels for a product, the information that they would contain, must be encrypted by the private key of the manufacturer. The smartphone application comes with the manufacture's public key. A successful decoding of the QR code using the smartphone application, therefore, ensures that packaging was done by the claimant manufacturer. To add more invincibility to this solution, the model suggests certain Algorithms that makes it more reliable without compromising much of the application's responsiveness.

The model prototyped and implemented by the authors can be placed in form of an Add-on firmware with the packing machinery. To broaden its scope, the thesis suggests both offline and online methods of the model. The next two sections describe these approaches.

2.1. Offline Verification Approach

In the offline verification approach, the manufacturer distributes a smartphone application which comes with the following knowledgebase

The public key of the manufacturer (K+M)

The asymmetric encryption technique using the public key of the manufacturer, viz. E+M(dataBytes [])

A mathematical function f(TS) that works upon the parameter, TS that represents a timestamp

A digest function θ(S) that works upon the parameter S, any String

A sequence of characters, *i.e.,* a String (DLE)

An operation, ∑DLE (A, B, C,…) that concatenates a number of Strings delimited with the given String (DLE)

The same knowledgebase is had with the software running on the manufacturer's packaging machinery. Details about these knowledgebase are described in [Table-1].

2.1.1. Generation of Product Specific QR-code for Labeling: At the end of the packaging machinery, the product information (S) and the current timestamp (TS) will be used to create the QR codes to be labelled with the products and the following steps will be carried out:

A pseudo-random number (R) will be generated

A value (PSK) will be calculated by taking the XOR of the random number to the value of the time-function, *i.e.,* f(TS)

PSK: = R⊕f(TS)

This value is referred to as the Product-Specific Key, because the probability of getting a duplicate value of P is very little.

The Theta-digest of the product information(S) is calculated. D: =θ(S)

The value of the Theta-digest is XOR-ed with the value the Product Specific Key.

D1: = D ⊕ PSK

The value of D1 is encrypted with manufacturer's private key to produce a valueD2; expressed as, D2:= E –M (D1)

Using DELIM as the delimiter, values of TS, R, D2 and S are concatenated using the concatenation operator to yield some text, Y

Y: = ∑ (TS, R, D2, S)

DELIM

A QR code is formed with the text, Y

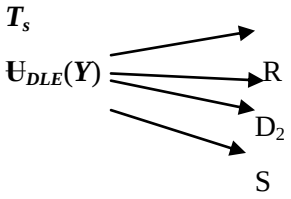
The QR code is labeled on the product or the package.

The steps above are carried out to generate product specific QR codes (labels), and are described in Algorithm 1 presented in [Table-2]. The same is presented as a flowchart as illustrated in [Figure-2].

2.1.2. Detection of a Spoofed Manufacturer: While making a purchase, the customer of the product uses her smartphone application that had been distributed by the original manufacturing brand. The application scans the **QR** code labelled on the packaging. A spoofed manufacturer identity will be always detected by the application and it will warn the customer about it. The functionality of the smartphone application is analyzed below:

a) The scanning of the **QR** code yields a text (**Y**)

b) This text is tokenized with the given delimiter (**DLE**), which should yield the timestamp ($\boldsymbol{T_s}$), a number (***R***), a number $\boldsymbol{D_2}$ and a String (***S***).



c) The value of the time-function is calculated.

$\mathbf{g} := f(T_s)$

d) The *Product Specific Key* (**P**) is obtained by XOR-*ing* the value of **R** with the value of the time-function calculated above.

$\mathbf{P} := \mathbf{g} \oplus \mathbf{R}$

e) A value $\boldsymbol{D_1}$ is obtained by taking *XOR* of the number $\mathbf{D_2}$with the value of the *Product Specific Key* (**P**)

$$\mathbf{D_1 := D_2 \oplus P}$$

f) $D_1$is decrypted with the public key of the manufacturer

$\mathbf{D} := \mathbf{E_{+M}}(\boldsymbol{D_1})$

g) The theta-digest is calculated for the product information (S)

$\mathbf{W} := \boldsymbol{\theta}(\boldsymbol{D})$

h) If $\boldsymbol{D == W}$, the identity of the claimant manufacturer is verified. Otherwise, a spoofing of the manufacturer's identity is sanguine.

The steps described above conform to *Algorithm 2* presented in **[Table-3]**.

2.2. Online Verification Approach

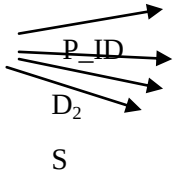
There is one major drawback to the previous approach. It fails if a fraud manufacturer copies some (or even, one) of the QR codes generated by the original manufacturer, in the labels of the fake products, the offline approach treats it as a genuine product. We must emphasize that this model never tests genuineness of a product (*e.g.*, a beverage or medicine); rather it aims to make an easy detection of a spoofed manufacturer identity from the label on the packet or bottle it comes in. The online approach described in this section addresses this loophole in the offline method discussed in **[2.1].**

2.2.1. Generation of Product Specific QR-code for Labeling: The steps of generating a product specific QR code in the online approach is almost same as in the offline approach except that the QR code contains a unique identifier for the product while the pair, pseudo random number and the product id are stored in the manufacturer's database. A web service can issue a database query to fetch the random number given a product id to verify if that is valid.The details of the steps of generating a product specific QR code is discussed in *Algorithm 3*, presented in **[Table-4]**.

2.2.2. Online Verification of the Manufacturer Identity: Before purchasing a product, the customer uses her smart phone application to decode the QR image involving the following steps:

a) The scanning of the **QR** code yields a text (**Y**)

b) This text is tokenized with the given delimiter (**DLE**), which should yield the unique product identifier (**P_ID**), the timestamp ($\boldsymbol{T_s}$), a number $\boldsymbol{D_2}$ and a String (***S***).



c) The application uses the product identifier (**P_ID**) to invoke a secure web service, provided by the original manufacturing brand. The web service queries the product database to fetch the ***PRN*** for the given product identifier (**P_ID**). The functionality of the web service is described in *Algorithm 5*, presented in **[Table-7]**. If the product identifier is a valid one, the web service should return the *Product Specific Random Number* (***PSRN***). If the web service returns a negative number indicative of invalid product identifier, the application warns the customer at once. Otherwise the steps below are carried out

d) The value of the time-function is calculated. $\mathbf{g} := \boldsymbol{f}(\boldsymbol{T_s})$

e) The *Product Specific Key* (**P**) is obtained by XOR-*ing* the value of **PSRN** with the value of the time-function calculated above.$\mathbf{P} := \mathbf{g} \oplus \boldsymbol{PSRN}$

f) A value $\boldsymbol{D_1}$ is obtained by taking *XOR* of the number $\mathbf{D_2}$with the value of the *Product Specific Key* (**P**); $\mathbf{D_1 := D_2 \oplus P}$

g) $\boldsymbol{D_1}$is decrypted with the public key of the manufacturer ;$\mathbf{D: = E_{+M}(}\boldsymbol{D_1}\mathbf{)}$

h) The theta-digest is calculated for the product information ($\boldsymbol{S}$)$\mathbf{W}$: = $\boldsymbol{\theta}$ ($\boldsymbol{S}$)

i) If **D==W**, the identity of the claimant manufacturer is verified. Otherwise, the customer is warned that the claimant manufacturing brand is not the original manufacturer.

The steps described above conform to *Algorithm 4* presented in **[Table-5]**.

This approach, however, is more reliable than the offline approach, though the customer must bear the data charges for accessing the web service over *Internet*. The web service also returns some information (E.g. a sequence of characters) which is unique for a product. The same information must be printed on the packet beside the QR code. Since the web service can be accessed by the smart phone application only (because it may be so developed to come with some digital signature which the web service entertains), the online verification process and the manual verification by the customer thereafter, guarantees that the product is unique.

2.3. Analysis of Reliability of the Online Verification Approach

We assume the following adverse situations that challenge the reliability of the proposed model:

a) A fake manufacturer X spoofs the identity of a reputed manufacturer brand, Y and creates random labelling on the packets of its products.

Solution: The smart phone application will detect that the product identifier is invalid

b) The fake manufacturer X itself purchases a genuine product by Y and clones multiple QR labels of the same product and tries these fake products.

Solution: This is a serious trouble, because the application logic will treat all the fake products labelled with one original QR code to be genuine. To address this issue, the original manufacturer (Y) must demand that a product is registered when it is sold. Y may levy certain business strategies like voiding the guarantee of a sold product if it is not registered within a span of few days (*e.g.,* 3 days). The retailer might also be contracted to register a product as soon as it is sold out. To register a sold product means that the unique product identifier will **cease** to be valid in the database. As another customer wants to buy a fake product labelled with a cloned QR label, the web service tells the smart phone application that this product identifier is no more valid. Thus, multiple clones of a QR code cannot mislead the verification process. We proposed this model with the assumption that the algorithm of generating product specific QR codes is available only to the firmware inside packaging machineries of the original manufacturing brand and that it can be decoded by only the smart phone application distributed by it through a secure, trusted party (*e.g.*, Google Play, Apple Store, Widows Store *etc.*,) we claim that this process can help customers detect a fake product with a forged packaging label very easily using their smartphones and can really turn out to be fruitful to prevent sale of adulterated medicines, health care and food products and save life of people. [**Figure-3**] shows some screen shots of an android implementation of the proposed model.

3. Tables

Table-1: Notations and symbols used in the thesis

1	$E_{+Entity}(D)$ $E_{-Entity}(D)$	An asymmetric cryptographic algorithm that encrypts Data (D) with the public key (suffixed with +) and private (suffixed with -)of an Entity (E.g. ***Server***)
		Parameters: D:= Data
		Corollary: $W=\mathbf{E}_{+Server}(D) \rightarrow D=\mathbf{E}_{-Server}(W)$
2	$\sum_{DLE}(A,B,C,...Z)$	Concatenate a number of objects (Texts or numbers) using a delimiter (*DLE*)
		Parameters: Any number of texts and/or numbers
		Implementation Algorithm: **Input:** *N*:= Number of Parameters *DLE*: =Delimiter *Params*[]: =Parameters Result:= EMPTY_STRING **FOR** J=0 **TO** (*N*-1) Result: = Result+ (*DLE* + *Params*[J]) **[X+Y means to concatenate Y to string X]** **NEXT** J **Output**= Result
3	QR(*aText*)	Create a QR code with a String (aText)
		Parameters:aText: = The text which will be the QR-coded.
4	QR^{-1} (*qrImg*)	Decode a QR code to extract the texts off it
		Parameters: qrImg:= A QR code image or byte stream
		Corollary: *qrImg*=**QR**(*someText*) →*someText*= $\mathbf{QR}^{-1}$ (*qrImg*)
5	$\cup_{DLE}(S)$	Extract the delimited tokens from a string (*S*) given a delimiter (*DLE*).
		Parameters: S:= a String DLE:= delimiter
		Corollary: $Y=\sum(A,B,C) \rightarrow \cup_{DLE}(Y) := \{A,B,C\}$ ***DLE***
6	$K_{+Entity}$ $K_{-Entity}$	The public and private keys of an entity E.g. $K_{+Server}$ means the public key of the Server; whereas, $K_{-Server}$ refers to its private key
7	$f(T_S)$	• This is a function of the timestamp (T_S) • This is a function which is available with the manufacturer's and with the smart phone application (distributed by the manufacturer to the customers) • This is referred to as the ***Time Function***
8	$\theta(S)$	• This is digest function that operates on a String. • $\theta(A) \neq \theta(B)$ ∀ $A \neq B$ • This digest function is available to the mobile application as well as the manufacturer side packaging machinery.

Table-2:
Algorithm 1: Formation of the QR code by the packaging machinery

	MANUFACTURER OR PACKAGER SIDE TECHNIQUE
1	**Input:** a) T_S := The current timestamp (Number) b) S := Product Information (String)
2	Create a pseudo random number **Outcome of this step:R:= *Rnd*()**
3	Compute the value of the *time function* **Outcome of this step: g**$:= f(T_S)$
4	Compute the product specific key (PSK) by *XOR-ing* the value of the time function with the pseudo random number generated in [Step-2] **Outcome of this step: PSK**$:= \mathbf{g} \oplus \mathbf{R}$
5	Calculate the digest value of the product information (S) using a digest algorithm (θ) **Outcome of this step: D: = θ (*S*)**
6	Take XOR of the value of this digest (D) with the Product specific key as the key. **Outcome of this step: D_1 : = D ⊕PSK**
7	Encrypt value calculated above, i.e. D_1 with the private key of the manufacturer/packaging plant (K_{-M}) **Outcome of this step:D_2 := E_{-M} (*D_1*)**
8	Create a text (***T***) with the concatenated values of the timestamp (***T_S***), the pseudo random number (***R***), D_2, and the product information (***S***). These are delimited by a predefined delimiter (***DELIM***) **Outcome of this step:*T*$:= \sum (T_S, R, D_2, S)$** **DLE**
9	Create a QR code Q with the text generated above. **Outcome of this step:Q**$:= QR(T)$
10	Get the QR code printed and labelled on the product's packet and/or product body. This is done by the packaging and labelling machinery. The Steps [1-9] are carried out by a software may be bundled with packing machinery as an add-on to the firmware or as a part of the *Real Time Operating System* running on the machinery.

Table-3:
Algorithm 2: Offline product verification method by a smart phone

	Input: **A. The label on the product**
1	The consumer uses her smartphone application to scan the QR code labelled on the packet of the product (E.g. a bottle containing a brand of cough syrups). The smartphone application extracts the text (T) off the QR code. **Outcome of this step:T**$:= QR^{-1}(QR)$
2	The text (T) is de-tokenized and yields its substrings *i.e.T_S , R, D_2 and S* **Outcome of this step:** **Z[]:= Ʉ$_{DLE}$ (*T*) , where T_S :=Z[0], R:= Z[1], D_2:=Z[2] S:=Z[3]**
3	Decrypt the value D_2 with the public key of the manufacturer (K_{+M}); This key comes with the smart phone application.

	Outcome of this step: $D_1 := E_{+M}(D_2)$
4	Calculate the value of the time function with the timestamp T_S **Outcome of this step: g**$:= f(T_S)$
5	Calculate the value of the product specific key ***(PSK)***. **Outcome of this step:PSK**$:=$ **g ⊕ R**
6	Obtain the value *D* by *XOR-ing* the value of D_1 with the product specific key (PSK). **Outcome of this step: D**$:=$ **D_1⊕ PSK**
7	Calculate the digest value of the product information (S) with the digest function θ. This common digest function, too, comes with the digest function. **Outcome of this step: W**:= $\theta(S)$

8	**IF** (*D*==*W*) **THEN**
	The product verification is successful
	ELSE
	The product verification failed
	END IF

Table 4.
Algorithm 3: Formation of product specific QR code at the manufacturer's end /packaging plant

Manufacturer Or Packager Side Technique	
1	**Input:** a) T_S := The current timestamp (Number) b) P_id:= Unique Product ID (An identifier string or number) c) S := Product Information (String)
2	Create a pseudo random number. **Outcome of this step:R:=** ***Rnd*()**
3	Compute the value of the *time function* **Outcome of this step:** **g**$:= f(T_S)$
4	Compute the product specific key (PSK) by *XOR-ing* the value of the time function with the pseudo random number generated in [Step-2] **Outcome of this step: PSK:= g⊕R**
5	Calculate the digest value of the product information (S) using a digest algorithm (θ) **Outcome of this step: D: =** θ **(*S*)**
6	Symmetrically encrypt the value of this digest (D) with the Product specific key as the key. **Outcome of this step:** **D_1 : = D ⊕ PSK**
7	Encrypt value calculated above, i.e. D_1 with the private key of the manufacturer/packaging plant (K_{-M}) **Outcome of this step:** **D_2 := E_{-M} (D_1)**
8	Create a text (T) with the concatenated values of the product identifier, the timestamp (T_S), D_2, and the product information (S). These are delimited by a predefined delimiter (DELIM) **Outcome of this step:** $T := \sum (P\_id, T_S, D_2, S)$ **DLE**

9	Create a QR code Q with the text generated above. **Outcome of this step:Q**$:= QR(T)$
10	A database containing a table-schema, as shown in **[Table-6]**, is maintained with the manufacturer (or the packaging center). Upon successful completion of [Step-9], the tuple (P_id, R) is inserted to the table in the database, as illustrated in **[Figure-1]**
11	Get the QR code printed and labelled on the product's packet and/or product body. This is done by the packaging and labelling machinery. The *Steps* [*1-10*] are carried out by a software may be bundled with packing machinery as an add-on to the firmware or as a part of the *Real Time Operating System* running on the machinery.

Table-5:
Algorithm 4: Online product verification by smart phone

Input: The QR code (Q) printed on the product or its package.	
VERIFICATION TECHNIQUE BY USER'S SMART PHONE	
1	The consumer uses her smartphone application to scan the QR code labelled on the packet of the product (E.g. a bottle containing a brand of cough syrups). The smartphone application extracts the text (T) off the QR code. **Outcome of this step:T**$:= QR^{-1}(QR)$
2	The text (***T***) is de-tokenized and yields the different parameters i.e. Product identifier (***p_id***) ,timestamp (T_S) , the crypto-digest (D_2) and the product information (***S***) **Outcome of this step:** **Z[]: =** $\mathbf{U_{DLE}(T)}$ **,** Where, **p_id:= Z[0]** $\mathbf{T_S}$**:=Z[1],** $\mathbf{D_2}$**:= Z[2]** **S:=Z[3]:**

3	The application avails the manufacturer's Web service over Internet (over HTTP *or* HTTPS) to query the **PRN** against the product id (p_id). The web service responds with the value (R) from the database shown in **[Table-6]**.
	The functionality of the web service is illustrated in *Algorithm 5***[Table-7]**
	Outcome of this step: The number**,** ***R***
	IF (*R==0***) THEN**
	• The product does not conform to the manufacture's database.
	• **Treat the product to be fake at once and stop execution at this step**
	ELSE
	• **GO TO Step-4**
	END IF

4	Decrypt the value D_2 with the public key of the manufacturer (K_{+M}); This key comes with the smart phone application. **Outcome of this step:** $\mathbf{D_1 := E_{+M}(D_2)}$
5	Calculate the value of the time function with the timestamp T_S **Outcome of this step: g**$:= f(T_S)$
6	Calculate the value of the product specific key *(PSK)* by *XOR-ing* the value of the *timefunction* to the value of **R** **Outcome of this step:PSK:= g ⊕ R**

7	By XOR-ing D_1 with the product specific key (***PSK***), we obtain the value, ***D*** **Outcome of this step: D:= $D_1 \oplus$ PSK**
8	Calculate the *digest* value of the product information (***S***) with the digest function θ. **Outcome of this step: W**:= $\theta(S)$

9	**IF** (D==W) **THEN**
	The product verification is successful
	ELSE
	The product verification failed
	
	END IF

Table-6: Schema of the database maintained with the manufacturer.

FIELD NAME	Type	CONSTRAINT	DESCRIPTION OF THE FIELD
Product_ID	Text/Number	**Primary Key**	The unique identifier of a product
PRN	Number		The pseudo random generated while executing *Algorithm 2***[Table-3]**

Table-7:
Algorithm 5: Functionality of the Web Service supporting online verification

Input: *An HTTP request*	
1	Extract the request parameter that corresponds to the product-id. E.g. String product_id=request.getParameter("p-id"); // A simplified Java Web implementation code-snippet **Outcome of this step:** the *product-id* (P) that has to be verified

2	Query the table, as shown in **[Table-6]**, from the backend database to obtain the value of the column PRN matching the product-id obtained in [Step-1].
	This can be accomplished by the simple pseudo-SQL:
	*Select***PRN***from****Prn_Table*** *where* **product_id**=*P*
	
	IF (NO_RECORDS_FOUND_FOR_PROUDCT-ID) THEN
	R:=0
	ELSE
	R:= Result Of (*Select***PRN***from****Prn_Table*** *where* **product_id**=*P*)
	END
	
	Outcome of this step: the number, ***R***

3	Create an HTTP response with the response body containing the value of ***R***

4. Figures

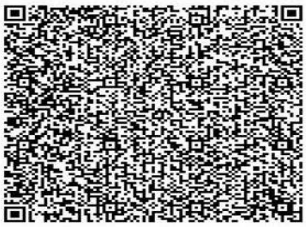


FIGURE-1: Product Specific PRN

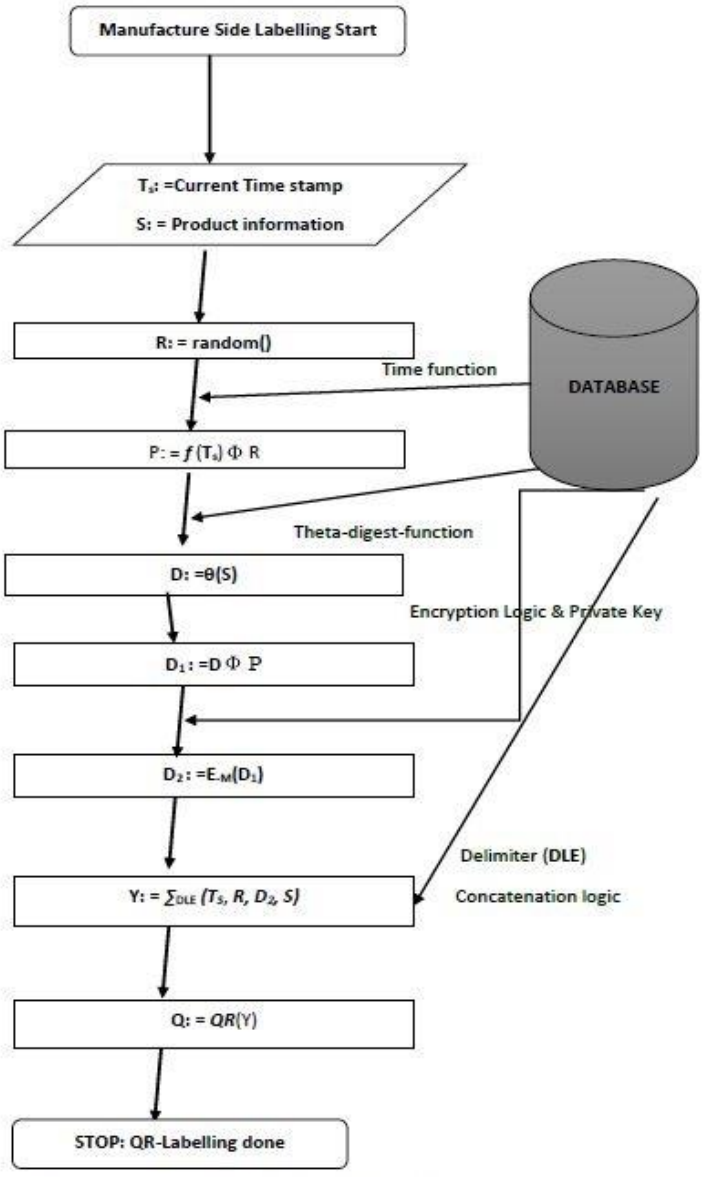


Figure-2: Generation of Product specific QR Code Labels

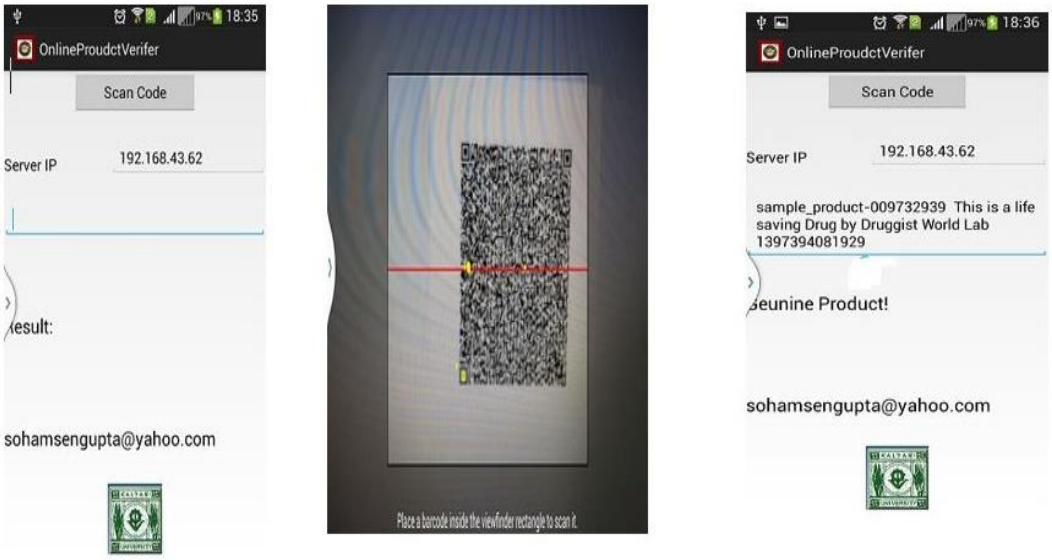


FIGURE-3:

ONLINE PRODUCT VERIFICATION BY A SMARTPHONE

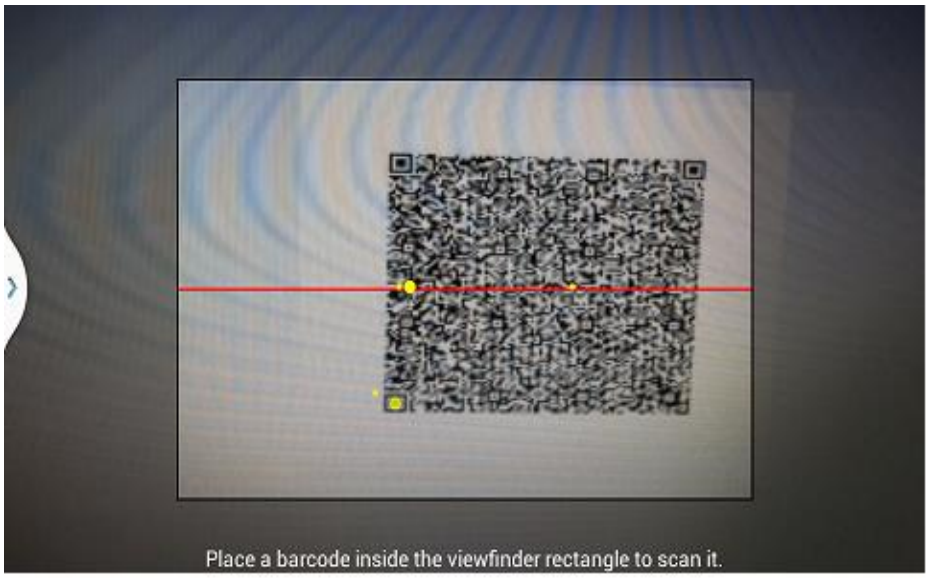


Figure 4: Screen Shot of Smart Phone assisted Verfiication

Figure 5: Screen shot of positive Verification Result

[M]

12216899437432354320353221393357543058879880691386487894847868579340021217749225854016137857292335155273033995385549964179143964553618679569609926197022108706653325561234849719785937795288414544989547319298446223034238571752533490772596258712871965333168157421395755610798386553578504306887194692326932592953 9

Exponent:

65537

FIGURE-6:

RSA PUBLIC KEY USED BY THE SMARTPHONE IN PRODUCT VEROFICATION

References

[1] J. F. Zandbelt, R. J. Hulsebosch, M. S. Bargh and R. Arends, "Trusted Directory Services for Secure Internet Connectivity: Transport Layer Security using DNSSEC", Electronic Notes in Theoretical Computer Science, vol. 197, Issue 2, **(2008)** February 22, pp. 91-103, ISSN 1571-0661.
[2] http://dx.doi.org/10.1016/j.entcs.2007.12.019.
[3] H. T. Panduranga, S. K. N. Kumar, "A Novel Image Encryption Technique using Multi-Wave Based Carrier Image, Procedia Engineering, vol. 38, **(2012)**, pp. 2998-3004, ISSN 1877-7058, http://dx.doi.org/10.1016/j.proeng.2012.06.350.

[4] H. B. Mahmood, "Transport layer security protocol in Telnet", Communications, 2003. APCC 2003. The 9th Asia-Pacific Conference on Computer Networks, vol. 3, DOI:10.1109/APCC.2003.1274255.
[5] V. L. Voydock and S. T. Kent, "Security mechanisms in a transport layer protocol, Computer Networks (1976)", vol. 8, Issues 5–6, **(1984)** October–December, pp. 433-449, ISSN 0376-5075, http://dx.doi.org/10.1016/0376-5075(84)90006-0.
[6] V. L. Voydock and S. T. Kent, "Security mechanisms in a transport layer protocol, Computers & Security", vol. 4, Issue 4, **(1985)** December, pp. 325-341, ISSN 0167-4048, http://dx.doi.org/10.1016/0167-4048(85)90051-3.
[7] S. Sengupta, "An approach to provide a network layer security model with QR code generated with shuffled GPS parameters as embedded keys traveling over Internet using existing IPv4 mechanism", Computer Networks, vol. 57, Issue 11, **(2013)** August 5, pp. 2313-2330.
[8] J. Jürjens, "A domain-specific language for cryptographic protocols based on streams", The Journal of Logic and Algebraic Programming, vol. 78, Issue 2, **(2009)** January, pp. 54-73, ISSN 1567-8326, http://dx.doi.org/10.1016/j.jlap.2008.08.006.
[9] A. Espejel-Trujillo, I. Castillo-Camacho, M. Nakano-Miyatake and H. Perez-Meana, "Identity Document Authentication Based on VSS and QR Codes", Procedia Technology, vol. 3, **(2012)**, pp. 241-250, ISSN 2212-0173, http://dx.doi.org/10.1016/j.protcy.2012.03.026.
[10] P. A. Shah, M. Yousaf, A. Qayyum and H. B. Hasbullah, "Performance comparison of end-to-end mobility management protocols for TCP", Journal of Network and Computer Applications, vol. 35, Issue 6, **(2012)** November, pp. 1657-1673, ISSN 1084-8045, http://dx.doi.org/10.1016/j.jnca.2012.05.002**.**
[11] "Schneier's Cryptography Classics Library" Applied Cryptography, Secrets and Lies, and Practical Cryptography**.**
[12] S. Gupta, S. Sengupta, M. Bhattacharyya, S. Chattrejee and B. S. Sharma, "Cellular phone based web authentication system using 3-D encryption technique under stochastic framework," Internet, 2009. AH-ICI 2009, First Asian Himalayas International Conference on, **(2009)** November 3-5, pp. 1-5, doi: 10.1109/AHICI.2009.5340322.

Authors

Soham Sengupta, he holds an M. Tech in the fieldof mobile computing and B. Tech (*Hons*. In Information Technology). He has an excellent academic and professional career till date.He has been in the field of teaching and research for last ten years. He has a strong hold ON the impelling technologies and is held inhigh esteem in the Industries. His research interest covers Java andother open sources, Object Oriented Design patterns, Clouds, RIA, Cross Platform Mobile Computing, Mobile Computing and communication, Bluetooth Applications and JSR-82, Computer Networks (Protocols, Security, Augmentation and Applications). He is an Android Exponent, also known for his proficiency in Java, RIA frameworks, Cross Platform Mobile Apps development and giving innovative cost effective solutions. Augmented Reality and Computer vision are his present passions. By profession, he is serving the department of Information Technology as an Assistant Professor at JIS College of Engineering, Kalyani, India; and he consulted by different industries as a Java System architect and Android Architect cum Device Integration Expert. Founder CTO of TECH IT easy Labs, he has a vast experience in Industries like IBM, TCS, Yotto Labs and Touchstone Tie up Private Limited etc. His passions include relating inter-disciplinary topics, and strongly denies a barrier of subjects or topics. Someof his contributions to open sources can be found at http://sourceforge.net/users/sohamsengupta, http://sohamsironline.weebly.com and the Google Android Market.

ParthaPratim Sarkar, he was felicitated with a Ph.D in engineering from *Jadavpur University* in the year 2002. He has obtained his M.E from Jadavpur University in the year 1994. He earned his B.E degree in Electronics and Telecommunication Engineering from Bengal Engineering College (Presently known as Bengal Engineering and Science University, Shibpur) in the year 1991. He is presently working as Professor at the Dept. of Engineering & Technological Studies, University of Kalyani. His area of research includes, Micro strip Antenna, Micro strip Filter, Frequency Selective Surfaces, and Artificial Neural Network. He has contributed to numerous research articles in various journals and conferences of repute. He is also a life Fellow of IETE and IE (India).