2016

Product Authentication Using Hash Chains and Printed QR Codes

Harshith R. Keni
South Dakota

Product Authentication Using Hash Chains and Printed QR Codes

By

Harshith R. Keni

A thesis submitted in partial fulfillment of the requirements for the

Master of Science

Major in Computer Science

South Dakota State University

2016

PRODUCT AUTHENTICATION USING HASH CHAINS AND PRINTED QR CODES

This thesis is approved as a creditable and independent investigation by a candidate for the Master of Science in Computer Science degree and is acceptable for meeting the thesis requirements for this degree. Acceptance of this does not imply that the conclusions reached by the candidates are necessarily the conclusions of the major department.

Manki Min, Ph.D.
Thesis Advisor Date

Steven Hietpas, Ph.D.
Head, EECS Date

Dean, Graduate School Date

This thesis is dedicated to my family, friends and teachers who have all helped me get to where I am.

Acknowledgements

I would first like to thank my thesis advisor Dr. Manki Min of the Department of Electrical Engineering and Computer Science at South Dakota State University. Dr. Min has always encourage open discussions and kept his door open for my questions. He consistently allowed my research to be my own work while continuously steering me in the right direction.

I would also like to thank Ms. Montana Earle who was part of the initial research team for this as part of her Research Experience for Undergraduates program at SDSU in Summer 2015. Her diligent efforts at studying various authentication algorithms lead to the foundation for this research effort.

Finally, I would like to that the Department of Computer Science and all the graduate faculty for the continuous support and encouragement I have received.

Contents

List of Tables

List of Figures

Abstract

Product Authentication Using Hash Chains and Printed QR Codes

Harshith R. Keni

2016

This thesis explores the usage of simple printed tags for authenticating products. Printed tags are a cheap alternative to RFID and other tag based systems and do not require specialized equipment. Due to the simplistic nature of such printed codes, many security issues like tag impersonation, server impersonation, reader impersonation, replay attacks and denial of service present in RFID based solutions need to be handled differently. An algorithm that utilizes hash chains to secure such simple tags while still keeping cost low is discussed. The security characteristics of this scheme as well as other product authentication schemes that use RFID tags are compared. Arguments for static tags being at least as secure as RFID tags is discussed. Finally, a scheme for combining RFID authentication with static tags to achieve security throughout the supply chain is discussed.

Chapter 1

Introduction

1.1 Need for Authentication

According to the International Trademark Association (INTA), counterfeiting is the illegal practice of manufacturing, importing/exporting, distributing, selling or otherwise dealing in goods, often of inferior quality [3]. It frequently describes both the forgeries of currency and documents, as well as the imitations of clothing, handbags, shoes, pharmaceuticals, aviation and automobile parts, watches, electronics (both parts and finished products), software, works of art, toys, movies. Counterfeit consumer products have a reputation for being sub-par, of a lower quality and sometimes even harmful to the user. This has resulted in the deaths of hundreds of thousands of people, due to automobile and aviation accidents, poisoning, or ceasing to take essential compounds (e.g. in the case a person takes non-working medicine) [7].

Pharmaceutical counterfeiting is especially alarming and is a serious problem in both developed and developing countries. Counterfeit drugs account for about 10% of the global market, amounting to losses around US $ 200 Billion [4]. Numerous cases of physical harm, and even death, have been reported from consumption of counterfeit medicine.

One solution that has been used to combat counterfeiting is detecting counterfeits. This usually involves using chemical or other scientific analysis to detect a counterfeit products. The other technique to combat counterfeiting involves preventing counterfeiting by verifying authenticity of the products. This is usually done using computer algorithms and is referred to as **Authentication**. These techniques include using both overt and covert tactics like holograms, tracers, taggants, inks and electronic tracking [6].

It is essential that packaging and labeling is engineered to reduce the risk of counterfeit goods or theft and resale of products [8, 16]. Counterfeit goods, unauthorized sales, tampering etc. can all be reduced by securing the packaging and labeling process. In fact, packaging can itself include security features such as RFID (Radio-Frequency Identification) tags, tracers or even specialized tracking hardware.

According to the Food and Drug Administration (FDA), electronic tracking methods are "the single most powerful tool available to secure the drug supply." [9]. Most electronic tracking methods employ some form of RFID-based authentication coupled with some form of unique identification codes.

Multiple schemes based on RFID have been explored to tackle authenticate products and identify counterfeits[14, 12, 15]. RFID technology requires using specialized hardware to read RFID tags and might not be feasible for space-constrained packaging like pharmaceuticals. Furthermore, they are expensive and are geared for use from the point of manufacture to the point of sale/despensing. It is not available at a per-package or per-pill granualarity for end-users [6].

Printing a static code on a product for authentication is relatively simple and needs less specialized equipment. Simple, widely available smartphone technology enables scanning static printed codes easily and is relatively cheap. Due to these reasons, an Android smartphone application that scans QR Codes (Quick Response Code) was thus chosen because of relative simplicity and ease of implementation. Additionally,

a printed QR code can potentially have a larger bandwidth of data transfer than an RFID tag[1, 2, 18]. Also, smartphones have simple HTTP (Hypertext Trafer Protocol) network communication capabilities enabling the implementation of messaging and security protocols, as well as the availability of multiple open-source libraries enabling rapid prototyping. Due to these reasons, this thesis proposes two schemes for product authentication that use these technologies which aim to achieve cheap, robust and secure product authentication for end-users.

1.2 Objectives

The main objective for this project was:

1. To devise a cost-effective electronic authentication solution
2. The solution must also be widely available easy to use for the general public
3. It must be as secure as existing algorithms that use RFID tags
4. A combined approach to tackle authentication along the supply chain must be investigated

1.3 Organization of Thesis

The rest of this thesis is organized as follows: Chapter 2 explores detailed workings of security algorithms as well as authentication schemes and aims to identify strengths and weaknesses in them. Chapter 3 describes a proposed scheme that aims to use static printed codes, like QR Codes, to authenticate individual products while addressing various associated security problems. Chapter 4 describes a proposed scheme that attempts to combine individual product and individual package authentication techniques to enable security and tracking along supply chains. Chapter 6

analyzes and evaluates the proposed schemes, as well as explores comparisons with other described schemes.

Chapter 2

Literature Review

2.1 Security background

2.1.1 Cryptographic hash functions

The basic building block for an authentication scheme involves some sort of cryptographic hash function. These cryptographic hash functions are mathematical algorithms that map data of arbitrary size to a bit string of a fixed size in a way which is impossible or infeasible to invert i.e. the only way to obtain the input data given some output data is to try a large number of inputs to see if a match is produced. One way hash functions are an extremely important tool in security applications and are widely used for multiple applications. The input data is usually referred to as the *message* and the output data is referred to as the *hash value*. Some popular cryptographic hash functions are MD5, SHA-1, SHA-2, etc.

An ideal cryptographic hash function has the following four properties [19]:

1. it computes the hash value for any given message very quickly
2. it is infeasible to generate a message from its hash value except by using a brute-force approach

3. any change to the message, even a single bit, changes the hash value extensively such that the new hash value does not appear related to the old hash value

4. collisions are infeasible and improbable i.e. two messages having the same hash value will not occur

2.1.2 Hash chains

Hash chains, which involve repeated application of a cryptographic hash function to a message, are based on Lamport's scheme [13] to securely authenticate a user a finite number of times using a one-way encryption function. This allowed the system to authenticate passwords even if an intruder could eavesdrop or tamper with the communication between the user and the system in a computationally efficient manner. A hash chain of length N is construcuted by applying a cryptographic hash function $h()$ recursively N times to an initial message s.

$$h^N(s) = h(h(...h(s)...))$$

Due to the one-way nature of cyptographic hash functions, $h^{N-1}(s)$ cannot be obtained even if $h^N(s)$ is known without knowing the initial message s. This key property of hash-chains is critical to security in product authentication, and multiple product authentication schemes make use of it.

2.1.3 Security and Privacy of RFID

The most common basic authentication scheme is RFID (Radio Frequency Indentification) product authentication. Generally speaking, these systems consist of RFID tags and RFID readers. The tags operate as transponders while the readers operate as transceivers [21]. For some more complex algorithms, a database server is also used to store metadata as well as information about both the tags and readers.

Due to the wireless nature of RFID tags, some fundamental security objectives such as confidentiality, integrity, availability, authenticaiton, authorization, non-repudiation and anonymity need to be designed into the system to achieve secure usage [11].

2.1.4 Security Objectives

Confidentiality

As the communication between reader and tag is unprotected in some cases, it is possible for eavesdroppers to listen in the immediate vicinity. Furthermore, the tag's memory can be read as well. Both these problems can be solved by mutual authentication.

Integrity

With a basic system, the integrity of the information on a tag is not guaranteed. The tag memory can also be manipulated. Using database servers to store information about tags with lookup tables helps in avoiding these issues. Furthermore, a hash chain or general hash function acts as a provision against random failures.

Availability

RFID systems can be disrupted by denial-of-service attacks by jamming the frequency used for communication or some other techniques. The algorithm trying to secure the system must account for such attacks. Introduction of a server component further adds another point for attackers to attempt such attacks. The server maintains some state-information about the tags, and as such might lose synchronization with the reader or the tag thereby hampering future authentication.

Authenticity

The authenticity of a tag is at risk since its memory may be copied and replicated onto other tags. Various measures need to be taken to make tags tamper resistant or to mitigate losses from such duplication.

Authorization

Unauthorized readers or readers which have been tampered with must not be able to falsely authenticate valid tags. Using secure distribution channels for readers as well as mutual authentication between reader and server (if one is used) can act as deterrents for such attacks.

2.1.5 Security Characteristics

Forward Security

Also referred to as *Backward Traceability*, a tag's future internal state must not be computable even if previous state was known [20]. This occurs if, given all the internal state of a target tag at time t, the attacker is able to identify target tag interactions that occurred at a time $t < t$ where t' is some time before t.

Anti-Denial of Service

Some denial of service is hardware dependent and can be unavoidable. The system design can prevent certain denial of service attacks where attempts to overload the server with authentication mistakes are made. The algorithm must be lightweight enough to prevent such attacks. Also, certain steps like handshaking can be taken to prevent loss of synchronization between the tags, readers and server.

Anti-Reply Attack

Simple capture of a valid authentication request and response and subsequent replay of the same request must not yield the same response. This can be achieved by maintaining the state of authentication as well as the current position along the hash-chain for any product in the server.

Anti-Insider Attack

Secure distribution of readers as well as mutual authentication between tags, readers and server help prevent insider attacks. If any single component is compromised through an insider attack, it will fail a mutual authentication step and avoid insider attacks. Using secure channels for distributing readers to end-users or making the readers available only through specific trusted sources ensure the authenticity of the reader itself.

Mutual Authentication

Mutual authentication enables secure communication through complex systems as well as enables various componenets to be assured of authenticity of the other components they might interact with.

2.2 Review of various authentication protocols

Yang [24] proposed a hash-based RFID mutual authentication protocol. In this scheme, the reader authenticates the tag, the tag authenticates the reader, the reader authenticates the database, and the database authenticates both the reader and the tag. Also, initial values are updated occasionally. A shared secret between the components allows each component to authenticate each other. Updating this secret is a way of creating forward security. Also, this protocol is equipped to deal with replay

attacks. However, this protocol can easily fall prey to denial of service attacks and insider attacks.

Tsudik [22] proposed a protocol to identify RFID tags untraceably, called YA-TRAP. Simply, the tag receives a message and checks that the time stamp with this message is after the previous time received and before the maximum that is set. The server validates the valid or invalid message from a look up table for this tag. But, this protocol is very susceptive to denial of service attacks against the tag.

Henrici et al. [10] described a protocol in RFID systems that implemented a hash-chain called the Triggered Hash Chains Approach - where the value in the RFID tag is updated only when triggered by the backend server. This approach uses the reader simply as a channel of communication between the tag and the server which mutually authenticate each other. Denial of service attacks are prevented by the system's ability to backtrack one step. Also, since different hashes are used on the ID and the messages, an insider attack is thwarted. There is some computation weight on the server but overall this is also a very non-complex protocol. There is no prevention against the fact that the hash chain will eventually loop which can result in creation of a counterfeit.

Cho et al. [5] designed a secret value hash-based mutual authentication protocol where Mutual Authentication between the tag and server occurs. This approach makes a strong assumption that the communication between the reader and server are over a secure channel which is not possible to always ensure.

Ohkubo et al. [18] proposed a hash-based authentication protocol. The aim of the protocol is to provide better protection of user privacy with the basic concept of refreshing the identifier of the tag each time it is queried by a reader. The protocol changes RFID identities on each read based on hash chains. The Hash chain method is used in these two ways communication of RFID tag. This protocol does not require

a random number generator. However, it is confirmed that this protocol is flawed to certain replay attacks [21].

Weis et al. [23] proposed using cryptographically controlled hash locks for RFID tags. They argued that while hash values can be read by any reader, only authorized ones would be able to look up the tag information from the server. The objective was to improve security and privacy by using an integrated hash function where key could be verified by the server. This protocol does not ensure forward security and is traceable.

Song et al. [20] created a RFID authentication protocol for low-cost tags. There are new random numbers between the server and tag which are XORed and bit-shifted to update secrets used each time, so replay attacks are prevented. Also, this scheme provides forward security because the new secret is created using both random numbers generated that round. Additionally, there is mutual authentication between tag and server. But, this protocol assumes that the communication between the reader and the server is secure, and the tag is susceptible to denial of service attacks. It does not check the initial random number received and thus can receive the same one repeatedly and try to keep up with its own calculations and responses and get overloaded.

In Tsudik's and Song's work[22, 20], and many of the others it is assumed that there is a secure channel of communication between server and reader. In Yang's work[24], it is discussed that most protocols make that assumption. Also, the Tsudik's server[22] is assumed not attackable without proper reasoning. In the protocol proposed in this paper, we make no such assumptions.

Chapter 3

Single Product Authentication

In this chapter, we describe our proposed scheme for authenticating a single product using static QR code based hash chains. First we are going to discuss the design consideration of our scheme, explain our scheme, and then analyze our scheme.

3.1 Design Consideration

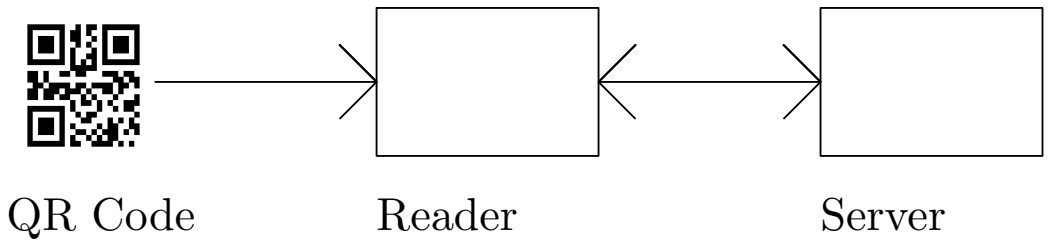

Figure 3.1: Communication Overview

Most RFID solutions for product authentication use dynamic RFID tags. Some computation and complexity is thus at the RFID end of the scheme. Our solution, which uses a statically printed code, does not have the luxury of having the tag performing any sort of computation as it just contains a hash value. The computational complexity is thus delegated to the reader and server. The first requirement our

scheme needed to have is to securely authenticate a product. Additionally, the number of times a product could be authenticated has to be chosen flexibly, depending on the application, to eventually prevent counterfeiting of the tag cloning. Finally our scheme must be available to the end-users for product authentication purposes through widely used distribution method. Our scheme is based on three entities: tags, tag readers, and the authentication server.

Since the tags have only one functionality (being read), the responsibility of the dynamic tags such as RFID to keep the secret secure has now moved to the mutual responsibility of the tag reader and the authentication server. In addition, it is important that the revelation of the static tag on the product does not reveal any other important secret and the hash-chain technique will ensure this security. Limiting the total number that a product can be authenticated can work in analogous to secret revocation in case of secret leak. Moreover, by printing unique static tag on each individual product item, we can minimize the risk of mass revocation. Let's say an attacker creates C counterfeits of an individual product P. The manufacturer of P has set N authentications as the limit and has made M such products. Assuming that care is taken such that $N \ll C$ is always the case, the manufacturer's loss for this product goes from $C/(C+M)$ to $N/(C+M)$ which is also a smaller number. Thus, the lifetime of C is limited and easier to identify. One situation that we can consider is the authentication attempt by many customers before it is being purchased. Some products may not be scanned as frequently as some others. In order to handle this issue flexibly we make the authentication lifetime dependent on both the product item and the tag reader. This way a product item can be authenticated beyond the authentication lifetime if it is being scanned by different readers.

We have to design a reader capable of reading the QR code, processing the security protocol and relaying the correct information to the server. Because the QR Code is easy to be read by any third party, the protocol had to securely authenticate the

read QR code between the reader and server. The message passing between the reader and server also had to be secured and authenticated to prevent replay and insider attacks. These can both be prevented by setting up a mutual authentication mechanism between in reader and the server. We achieved this by using a secret key that is hardcoded at both ends which is XORed and digested using a cryptographic hash function similar to the protocol described in [24]. This method allows us to generate communication tokens efficiently at both ends which can be verified and updated.

3.2 Algorithm

In this section, we are describing our algorithm depicted in Figure. 3.2 and Figure. 3.3 in the perspective of the reader R_{id} and the product P_{id}. The notations used in our algorithm are summarized as below.

h	A cryptographic hash function
f	Another hash function,
P_{id}	A unique id for the product
R_{id}	A unique id for the reader
S_s	A secret key shared with R_{id} stored at the server
S_r	A secret key shared with the server stored at R_{id}
N	Maximum number of allowed authentications
a	Number of times P_{id} has been authenticated
v	A hard-coded secret securely stored on the reader and server
k_n	n^{th} value in the hash chain for P_{id}, $k_n = h^{n-1}(k_1)$
M	Authentication message
$\oplus$	XOR operator
$\|\|$	Concatenation operator
$\leftarrow$	Substitution operator

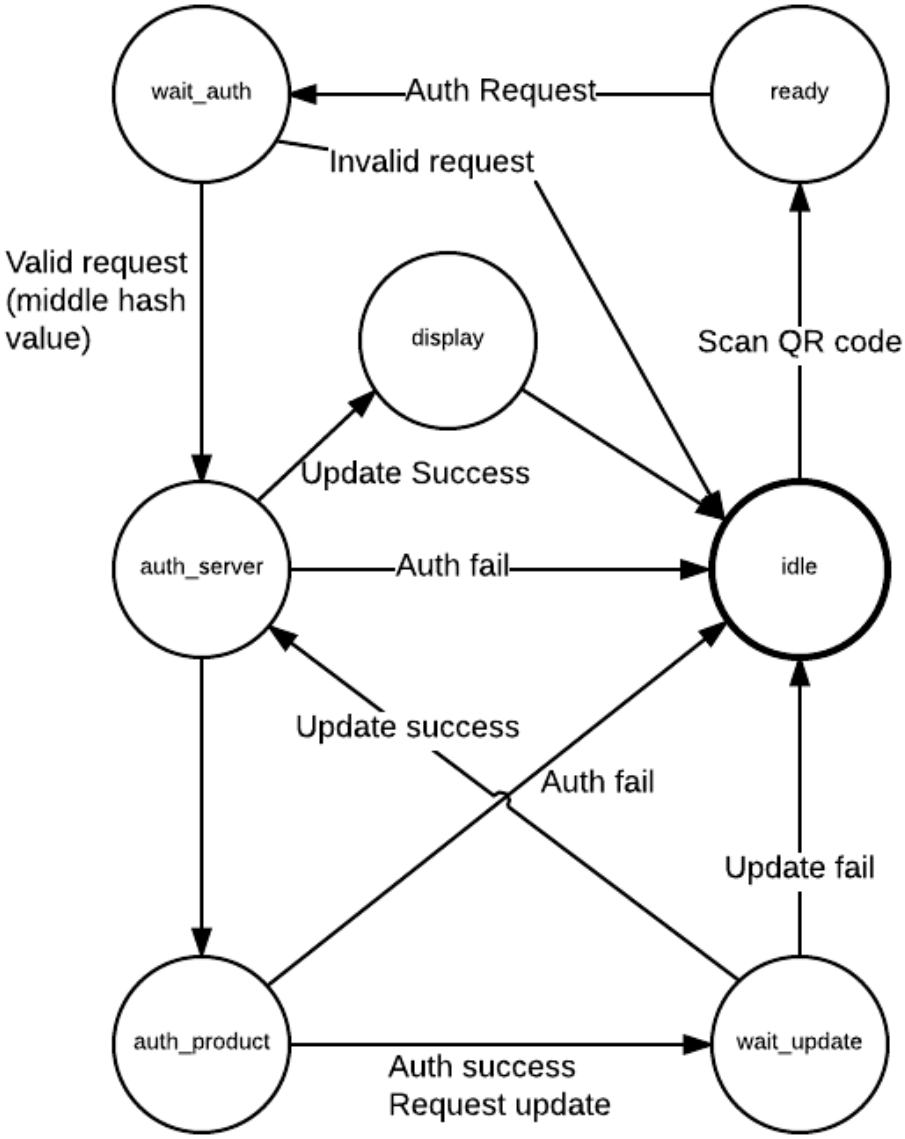

Figure 3.2: Reader Communcation Overview

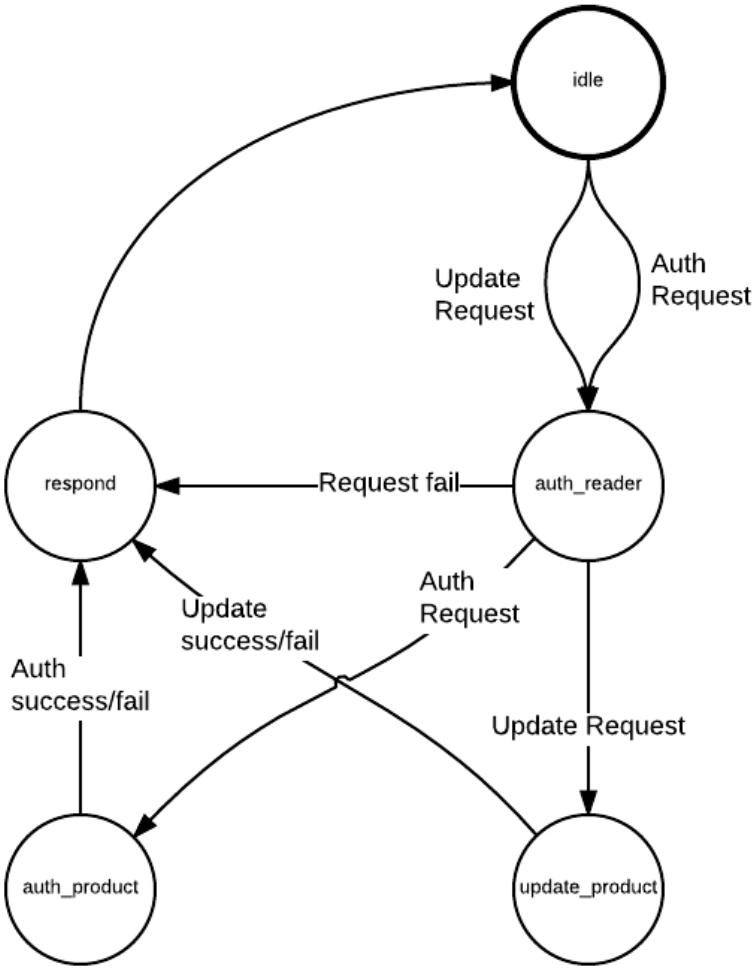

Figure 3.3: Server Communcation Overview

3.2.1 Reader (idle)

Wait for user to scan

In this state, the reader is idle and waiting for the user to scan a QR code.

3.2.2 Reader (ready)

read $P_{id}||k_{N+1}$

$t \leftarrow h(S_r)$

$s \leftarrow h(t)$

$M_r \leftarrow s||R_{id}||P_{id}$

send M_r

The user scans a QR code with the reader. The reader sends a message to the server with its twice-hashed secret, the Reader ID and the Product ID. The readers secret is used for mutual authentication with the server.

3.2.3 Reader (auth_server)

if $t' = t$ **then**

// Server is authenticated

$S_r \leftarrow f(S_r \oplus v)$

if $ValidAuthRequest$ **then**

goto auth_product

else if $UpdateSuccess$ **then**

goto display

end if

$s \leftarrow h(h(S_r))$

return $M_r||s||R_id||P_id'$

else

// Server is not authenticated

display error

end if

This is an intermediate step for authenticating the server as well as acknowledging the received message by the reader.

3.2.4 Reader (auth_product)

if $M_s \neq$ error **then**

$k' \leftarrow h^a(M_s)$

end if

if $k' = k_{N+1}$ **then**

// Product is authenticated

$M_r \leftarrow$ valid

display Authentication success

else

$M_r \leftarrow$ invalid

display Invalid product

end if

The reader will calculate a middle hash value here and compare with the one it received from the server. If they match for the same number of hashing steps, then we know that the product has been authenticated.

3.2.5 Server (auth_reader)

$t' \leftarrow h(S_s)$

$s' \leftarrow h(t')$

if $s' = s$ **then**

// Reader is authenticated

if *AuthRequest* **then**

goto auth_product

else if *UpdateRequest* **then**

goto update_product

end if

$S_s \leftarrow f(S_s \oplus v)$

return $M_s||a||t'$

else

// Reader is not authenticated

$M_s \leftarrow$ reader auth fail

return M_s

end if

This is an intermediate step for authenticating the reader as well as looking up or updating authentication records for a given Product ID. It also responds with the server-side mutual authentication information for the reader to authenticate the server.

3.2.6 Server (auth_product)

lookup R_{id}, P_{id}

if found **then**

lookup k_1, a

if $a > N$ **then**

$M_s \leftarrow$ product auth fail

else

$M_s \leftarrow k_{N-a}$

end if

else

$M_s \leftarrow$ product auth fail

end if

The server will lookup the product as well as the number of authentications. This step ensures that the lifetime of a product being authenticated is limited by a predefined maximum authentications. A middle hash value in the chain is computed based on the number of previous authentications and returned back to the reader.

3.2.7 Server (update_product)

if $M_r =$ valid **then**

$a \leftarrow a + 1$

end if

The server will update the number of authentications in a separate step after a handshake in order to ensure the synchronization of the authentication information of a given product in the database.

3.3 Security Characteristics

There are many ways for an attacker to go after a security protocol. We have done our best to try and think of them and come up with the best way to counteract these attacks. Of course, no protocol is one hundred percent secure. The goal of our approach is to address all possible security issues present in the previously described protocols while maintaining the simplicity of being able to authenticate a product using just a static QR Code.

First, to limit the lifetime of counterfeits, the number of authentications through the hash chains were made finite to a maximum number N through an established contract between the reader and server. The exact number N can be modified based

on the actual application. The first value of a hash chain is always a randomly generated unique string for each individual product which is securely stored on the server when the product is registered with the server. The hash-chain for that product is also calculated at this time and a QR code the last hash value in the chain along with its unique product ID is generated. This process can easily be integrated into the product's package printing process.

Because the production of QR codes is inexpensive, a motivated attacker might obtain multiple valid products and with enough understanding or reverse engineering of the system, might create a few counterfeits of each QR code to create a large number of counterfeit products. As the intention of this scheme is for authenticating cheap, mass-produced items, this kind of effort might not be worthwhile for an attacker for the relative profits of such counterfeits. Furthermore, it will be physically time-consuming to obtain enough products with unique QR codes and individually copying each one. Some techniques like special inks that only work in a specific lighting condition that is made available in a retail store for consumers to check validity can also be used to combat such attacks.

The first time a reader is used, it registers its unique ID with the server and the first shared secret is generated at both ends using this ID. Through the use of shared secrets that update at each message exchange, the server and the reader mutually authenticate each other. Also, the update process of the secrets, and the use of the hash chain to check hash provides forward security. The mutual authentication and update of the secret prevents a fake reader and a fake server. Therefore, insider attacks are prevented through this protocol. The update of the secret also prevents a replay attack. The messages change for each authentication and thus the same one cannot be reused. Because the QR Code is just a statically printed code, denial of service attacks against it are trivially impossible. Denial of service attacks against

the server can be prevented through various security measures like throttling and whitelisting/blacklisting IP addresses.

Also, a concern is a fake reader that impersonates the real reader and simply does not communicate with the server. The assumption we made here was that the distribution channel - like an app market - handles the authenticity and security of such apps. Impersonating a reader that communicates with the server is impossible as they mutually authenticate each other. Also, it can be assumed that end-users who are conscious of product authentication are also inclined to only use authentic readers from the app market.

3.3.1 Forward Security

Forward security is ensured by the use of hash-chains. As only a middle value between the first and last value in the hash-chain, going backwards from the last value to the first, is ever revealed and this cannot be traced back due to the properties of cryptographic hash functions, forward security is maintained.

3.3.2 Denial of Service (DoS)

Denial of service at the QR Code tag level involves obstruction/destruction of the tag which is trivial. This type of attack is not very beneficial to the attacker and only results in the authentication being denied. This is not preventable but the physical nature of the code means it is localized and difficult to damage codes on a large scale.

At the reader level, the application should be distributed through a secure channel like an app store. Thus, the app itself should be securely installed. Denial of service could involve disabling the camera or the smartphone itself, which is prevented by good design of the smartphone and is outside the scope of this project. No massive DoS attacks are practical to QR Codes or the reader app.

At the server level, as it is a HTTP(S) server, multiple types of denial of service attacks can occur. There are multiple research projects to address this and well known techniques exist to prevent denial of service for web servers. Also, the server operates in multiple authentication steps which involves lightweight computation such as just two hash functions. This eases the burden of handling DoS attacks on the server side.

3.3.3 Replay Attack

The lifetime of each Product ID and thus, each QR code is limited by the length of the hash chain. It can only be authenticated a finite number of times. Thus, it is infeasible to make a large number of a copies of a single QR code. This discourages copying (replaying) of QR codes.

There is mutual authentication between the reader and server using shared secrets when the reader requests to authenticate a certain code. As each message has a unique token depending on the current secret, each message is guaranteed to be unique. Thus, replay attacks can be prevented between the reader and server.

3.3.4 Insider Attack

The QR codes, being static, are trivially unaffacted by insider attacks.

The reader app needs to be distributed through a trusted and secure distribution channel like an app store to prevent insider attacks. Even if an attacker clones the reader app, because of the initial reader registration process and shared secrets and secret resynchronization, only a single clone will ever be validated at any given time. Multiple invalid requests from the same reader ID can be monitored and blocked at the server side. Furthermore, the mobile operating system should ensure that the app is sufficiently sandboxed and its inner functioning obfuscated to prevent intentional or accidental third party access.

The server will rely on standard web server techniques for prevention of insider attacks.

3.3.5 Mutual Authentication

As the QR Code itself is naive and unable to perform any computation, it cannot perform any sort of mutual authentication with the reader. Instead, the server will authenticate the QR Code. The reader and server mutually authenticate each other instead for an added layer of security.

3.4 Complexity Analysis

The following section analysis the performance complexity of the algorithm. A key to the notation used in analyzing the algorithm as well as for comparison with other algorithms is as follows.

H	A single computation of a crpytographic hash function
S	Subtraction computation
R	A random number generation computation
K	A symmetric encryption/decryption computation
L	128-bit variable stored
N	Number of tags stored in database
M	Modulo operation
X	XOR operation
i	Rounding operation
a	Number of authentications

3.4.1 Reader (idle)

Wait for user to scan

3.4.2 Reader (ready)

read $P_{id}||k_{N+1}$ *Space:* L

$t \leftarrow h(S_r)$ *Time: H, Space: L*

$s \leftarrow h(t)$ *Time: H*

$M_r \leftarrow s||R_{id}||P_{id}$

send M_r

3.4.3 Reader (auth_server)

if $t' = t$ **then**

// Server is authenticated

$S_r \leftarrow f(S_r \oplus v)$ *Time: H + X, Space: L*

if $ValidAuthRequest$ **then**

goto auth_product

else if $UpdateSuccess$ **then**

goto display

end if

$s \leftarrow h(h(S_r))$ *Time: 2H, Space: L*

return $M_r||s||R_id||P_id'$

else

// Server is not authenticated

display error

end if

3.4.4 Reader (auth_product)

if $M_s \neq$ error **then**

$k' \leftarrow h^a(M_s)$ *Time: H * a, Space: L*

end if

if $k' = k_{N+1}$ **then**

// Product is authenticated

$M_r \leftarrow$ valid

display Authentication success

else

$M_r \leftarrow$ invalid

display Invalid product

end if

3.4.5 Server (auth_reader)

$t' \leftarrow h(S_s)$ *Time: H, Space: L*

$s' \leftarrow h(t')$ *Time: H, Space: L*

if $s' = s$ **then**

// Reader is authenticated

if *AuthRequest* **then**

goto auth_product

else if *UpdateRequest* **then**

goto update_product

end if

$S_s \leftarrow f(S_s \oplus v)$ *Time: H + X, Space: L*

return $M_s||a||t'$

else

// Reader is not authenticated

$M_s \leftarrow$ reader auth fail

return M_s

end if

3.4.6 Server (auth_product)

lookup R_{id}, P_{id} *Space: R, N*

if found **then**

lookup k_1, a *Space: a*

if $a > N$ **then**

$M_s \leftarrow$ product auth fail *Space: L*

else

$M_s \leftarrow k_{N-a}$ *Time: H(N - a), Space: L*

end if

else

$M_s \leftarrow$ product auth fail

end if

3.4.7 Server (update_product)

if $M_r =$ valid **then**

$a \leftarrow a + 1$

end if

3.5 Comparison

Table. 3.1 shows a comparison of the security characteristics of our scheme to various product authentication schemes that use RFID. As discussed previously, these security characteristics are the basis of calling an algorithm secure. It is necessary that all of these must be satisfied to prevent or resist major security attacks and achieve all the security objectives discussed in Sec. 2.1.4. As it can be seen, QR Code Authentication indeed does satisfy these objectives and has the necessary security characteristics for good security. This is achieved through clever use of hash-chains for authenticating tags, mutual authentication between the reader and server and using lightweight, secure algorithms and platforms in the system.

Table 3.1: Comparison of Security Characteristics

Authentication Protocol	Forward Security	Anti-Denial of Service	Anti-Replay attack	Anti-insider attack	Mutual Authentication
Hash-based RFID Mutual Authentication Protocol [24]	Yes	No	Yes	No	Yes
YA-TRAP[22]	No	No	Yes	No	No
Triggered Hash Chain[10]	No	Yes	No	No	Yes
Secret Value Hash-based Mutual Authentication[5]	No	No	No	No	Yes
Cryptographic Approach to "Privacy-Friendly Tags" [18]	No	No	No	No	No
Hash-Based Access Control [23]	No	Yes	Yes	No	No
RFID-Authentication for Low-Cost Tags [20]	No	No	Yes	Yes	Yes
Our Protocol	Yes	Yes	Yes	Yes	Yes

A Comparison of the complexity of the algorithms used can be seen in Table. 3.2. Each algorithm is evaluated using the original provided algorithm by its author(s). The methodology used is similar i.e. counting the presence of various types of key computations, accounting for the various different techniques and notations used to present the algorithms.

Table 3.2: Complexity Analysis

Authentication Protocol	Time complexity	Space complexity
Hash-based RFID Mutual Authentication Protocol [24]	Tag: $4H + 1R$ Reader: $5H + 2R + 1K$ Database: $3H, 1K$	Tag: $1L$ Reader: $1L$ Database: $2NL$
YA-TRAP [22]	Tag: $1H + 1S + 2C$ Reader: - Database: L	Tag: $3L$ Reader: clock Database: NL
Triggered Hash Chain [10]	Tag: $3H$ Reader: - Database: $3H$	Tag: $1L$ Reader: - Database: $5NL$
Secret Value Hash-based Mutual Authentication [5]	Tag: $2H + 2M + 1R$ Reader: R Database: $2N$ (worst-case)	Tag: $2L$ Reader: - Database: $3NL$
Cryptographic Approach to "Privacy-Friendly Tags" [18]	Tag: $2H$ Reader: - Database: $2NiH$	Tag: $1L$ Reader: - Database: $2NL$
RFID-Authentication for Lot-Cost Tags [20]	Tag: $3H$ Reader: $1R$ Database: $2NH, 1H$	Tag: $1L$ Reader: - Database: $4NL$
Our Protocol	Tag: - Reader: $(5+a)H+X$ Database: $(4 + N - a)H + X$	Tag: - Reader: $5L$ Database: $(R + N + 4)L + N * a$

Chapter 4

Multiple Product Authentication

In this chapter, we describe our proposed scheme for authenticating a product along with its parent packaging using static QR code based hash chains. First we are going to discuss the design consideration of our scheme, explain our scheme, and then analyze our scheme.

4.1 Design Consideration

For this combined approach, both the packaging and inner products must be autheticated together. This is required to track a product through its supply chain and authenticate it along the way through the whole supply chain. Also, it is assumed that multiple levels of packaging can exist in the supply chain and each outer layer can have multiple products inside. The basic case considered here is two layers, a packaging layer which contains multiple products, both of which must be authenticated.

The combined authentication scheme must be able to maintain this hierarchical relationship between products and packaging, as well as track the progress through the supply chain while authenticating at each point.

This approach to achieve these goals include combining RFID based authentication at the package level with the QR Code based scheme at the product level. As the packaging needs to only be authenticated at the supply chain level, and not by the end user, the costs associated with obtaining an RFID reader can be justified for the added layer of security. Furthermore, this allows independent authentication using QR Codes for individual products as described in Ch. 3. A simple RFID scheme like the ones described in [22, 5, 18, 24] or the scheme described in Ch. 3 can be used for package authentication with a few modifications. This algorithm attempts to combine the expensive but secure RFID based authentication at the supply side with cheaper QR-Code based authentication at the user side.

The primary requirement is to maintain the hierarchy of products on the database server. It is also a good idea to record the authentications in database server for tracking purposes. An additional and optional locking step can also be included in child products to enable theft detection.

4.2 Algorithm

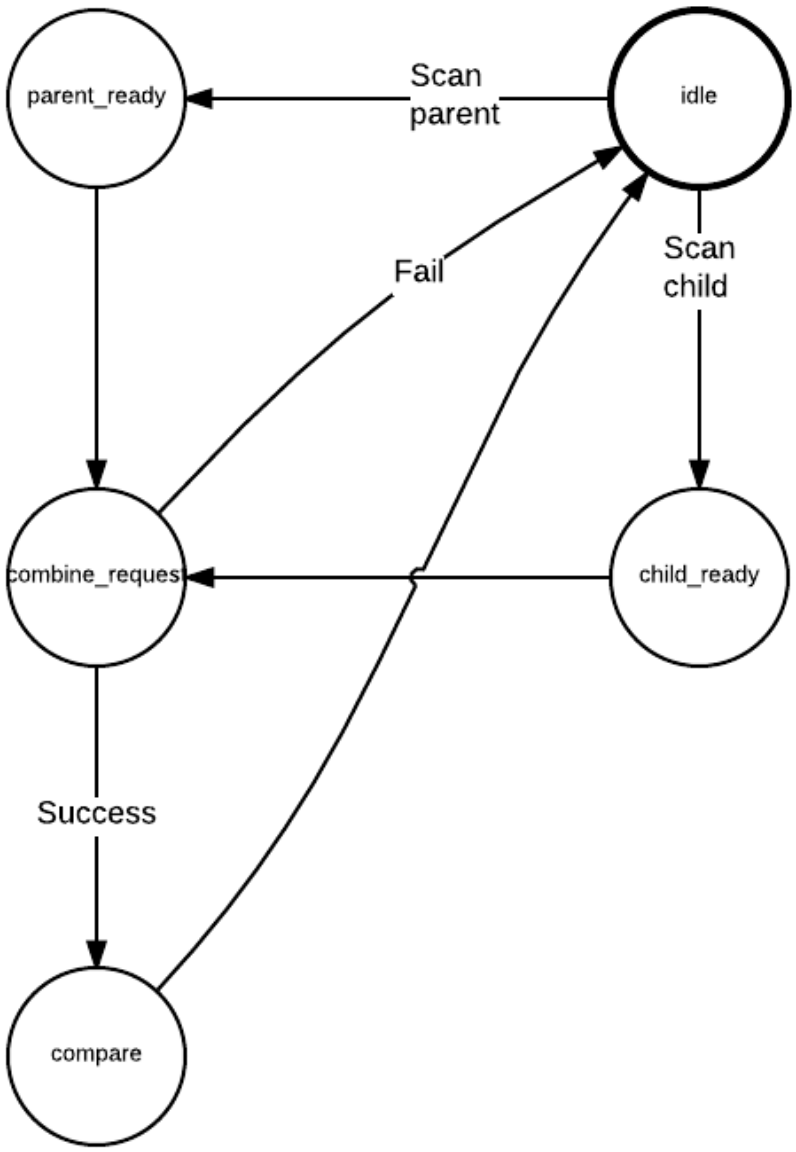

Figure 4.1: Communication Overview

In this section, we are describing our algorithm depicted in Figure. 4.1 The notations used in our algorithm are summarized as below.

f	A cryptographic hash function
PP_{id}	A unique id for a parent (outer) packaging
CP_{id}	A unique id for a child (inner) packaging
S_s	A secret key shared with R_{id} stored at the server
S_r	A secret key shared with the server stored at R_{id}
k_n	n^{th} value in the hash chain for P_{id}, $k_n = h^{n-1}(k_1)$
M	Authentication message
$\|\|$	Concatenation operator
$\leftarrow$	Substitution operator

4.2.1 Parent Ready (parent_ready)

read $PP_{id}||PPk_n$

A user will scan the Parent tag. This includes information about the product id as well as a middle value in a hash chain obtained using some authentication algorithm that uses hash chains for security. It is assumed that this tag is authenticated already at the scan.

4.2.2 Child Ready (child_ready)

read $CP_{id}||CPk_{N+1}$

A user subsequently must also scan a Child tag. This includes the product id and authentication information similar to the protocol described in Ch. 3.

4.2.3 Combine (combine_request)

$M_r \leftarrow CP_{id}||PP_{id}||R_id$

send M_r

A combination of the read information along with the reader's unique id is sent to the server.

4.2.4 Compare (compare)

if $M_s = f(S_r||PPk_n||CPk_{N+1})$ **then**

display Auth success

else

display Auth failure

end if

The server must authenticate the reader and subsequently ensure that the authenticated parent tag is a valid parent of the scanned child tag. It must then authenticate the child tag and respond with an appropriate message. The reader must then authenticate the server and display the returned message.

4.3 Security Characteristics

As seen in the algorithm, the authentication of both the package and the product involves both the reader and the server, as well as the tags themselves. The RFID tag on the package will authenticate itself and relay its unique identifier to the reader. Similarly, the read will scan the QR code on the product to obtain its unique identifier. Then the server and reader will apply the same cryptographic hash function on the combination of the read data along with the device secret to assess authenticity. Furthermore, the server is responsible for checking the validity of the hierarchy that is passed by the reader. There exists mutual authentication between the reader and server as described in Ch. 3.

Like the protocol described in Ch. 3, this algorithm must ensure that it serves as a deterrent for as many types of attacks as possible. It must do this while being able to accomodate various algorithms for authenticating the packaging at the distributor or supply side as well as products inside that packaging at the user or consumer side.

First and foremost, the inclusion of a hierarchical, tree-like, relationship between various tags that are created by the system enables the tight coupling of the tags which allows for tracking as well as detecting unexpected authentication attempts. To ensure this, each tag must have unique identifiers as well as a securely and randomly generated hash chain. The number of possible authentications for parent and child

tags can be adjusted according to the actual application and nature of the supply chain.

Due to this tight coupling, if some information like the place, time and person that authenticates parent tags is captured within the system, tracking of any subsequent false-authentications, detection of counterfeits due to repeated false authentication etc. can be tracked quite easily and isolated to the point of loss in the supply chain. Furthermore, if we encounter child tags that are attempted for authentication before its parent is authenticated, we can easily detect a possible scenario of theft or some other form of loss in the supply chain.

As previously discussed, the reader and server must mutually authenticate each other to ensure that line of communication is secured to prevent various attacks.

4.3.1 Forward Security

Both parent and child tags have hash-chains associated with them. Based on previous arguments, it can be assumed that forward security is maintained at both points. Furthermore, because of the tight coupling of the tags, it is not possible to authenticate a child without its parent being authenticated. This step would involve actually opening the parent packaging physically to access the inner products. This ensures that authentication attempts at inner products must succeed a valid authentication of its parent as well as a combined authentication of both upon the package being opened.

4.3.2 Denial of Service

Denial of Service for QR code or other printed tags have been discussed previously. Denial of Service for RFID tags are possible over long ranges by frequency through various means including frequency jamming, electromagnetic pulses etc. The RFID

tags themselves must be able to resist such attacks and are outside the scope of an algorithmic approach to resist such attacks.

4.3.3 Replay Attack

Due to the use of hash-chains, the lifetimes of both the outer packaging and inner products are limited. Furthermore, the middle hash value that is used to authenticate the combination of these tags are always a different pair that are backwards untraceable. As the state of current authentication is held in a central database server, replay attacks are meaningless a certain pair of authentication hash values expire immediately after a single use.

4.3.4 Insider Attack

RFID tags are indeed capable of being vulnerable to insider attacks at an individual level. Overall, such tampering would only invalidate the RFID tag as the central database server stores state information for the tag and limits the lifetime of the number of authentications. Any insider attack at the RFID tag will thus only render that tag unable to be authenticated and will not lead to false-positive authentications.

4.3.5 Mutual Authentication

The mutual authentication between the tag and the reader is possible and dependent on the algorithm being used for authenticating the tags.

Chapter 5

Implementation

As an exercise to verify the algorithms through a proof of concept as well as to simulate its operation, we implemented the system using some basic technology.

The single product authentication algorithm was implemented on the Android smartphone operating system as the reader and a simple HTTP server. The camera present on smartphones were used to scan the QR codes. MD5 was the chosen cryptographic hash function used for this, though others could be used as well.

Each product had to be registered in the server prior to an online generation of the QR code for it, which could be subsequently printed. Furthermore, the reader devices would register themselves when first opened with the server if an internet connection was present. The server was hosted on the cloud for easy access across networks, but could be hosted on a local network as well.

The Fig. 5.1, Fig. 5.2 and Fig. 5.3 are screenshots that show the QR code being scanned as well as successful and unsuccessful authentications.

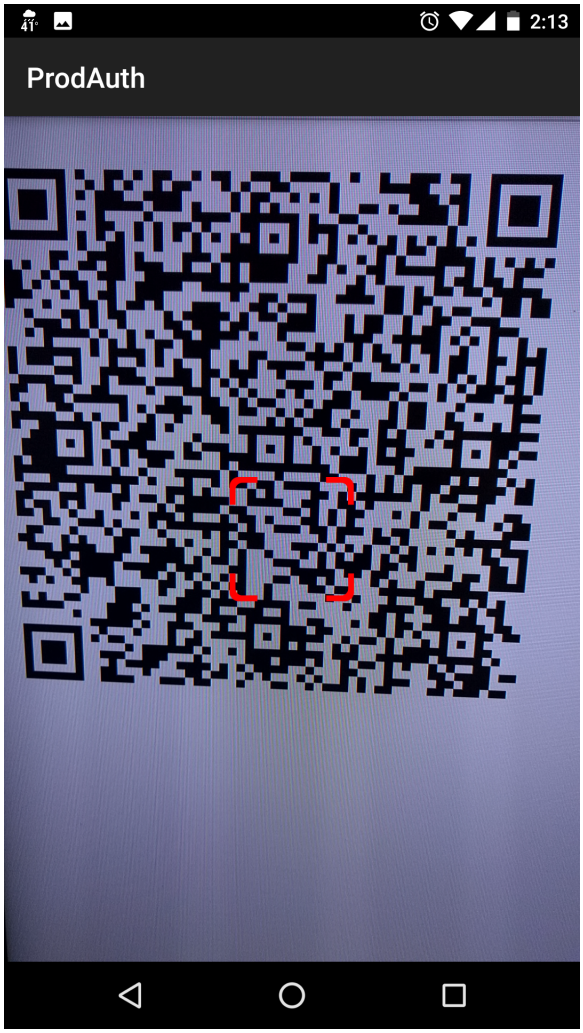

Figure 5.1: Scanning a QR Code

The multi-level authentication was simulated using a hash-chain for the parent packaging. The server built the hierarchy tree of products during the initial registration. It was assumed that every authentication of the parent packaging was successful and returned a middle hash value to the reader. The authentication was handled as described in Ch. 4 from there. The Fig. 5.4 is a screenshot of a parent and child packaging being authenticated together.

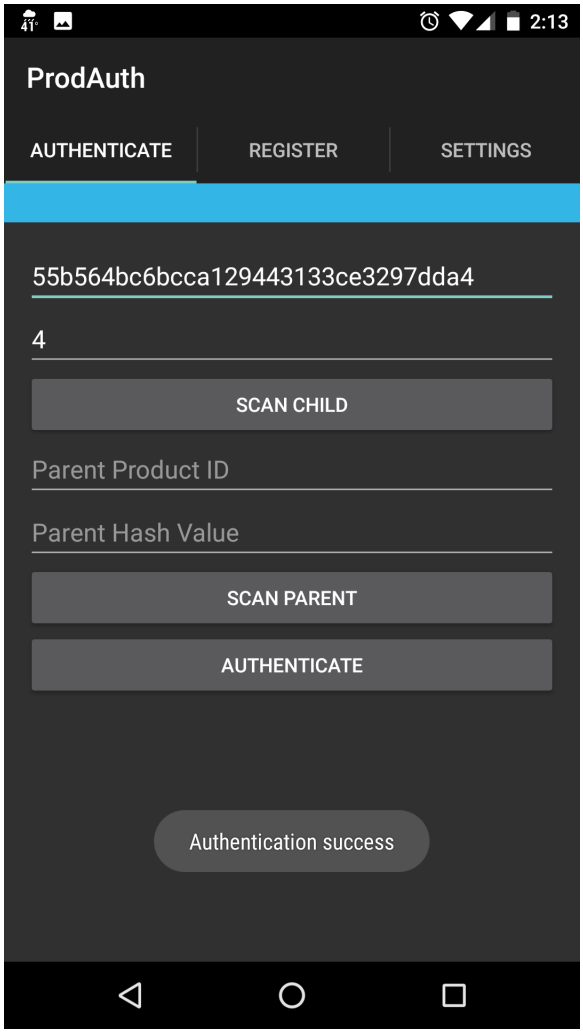

Figure 5.2: Successful Authentication

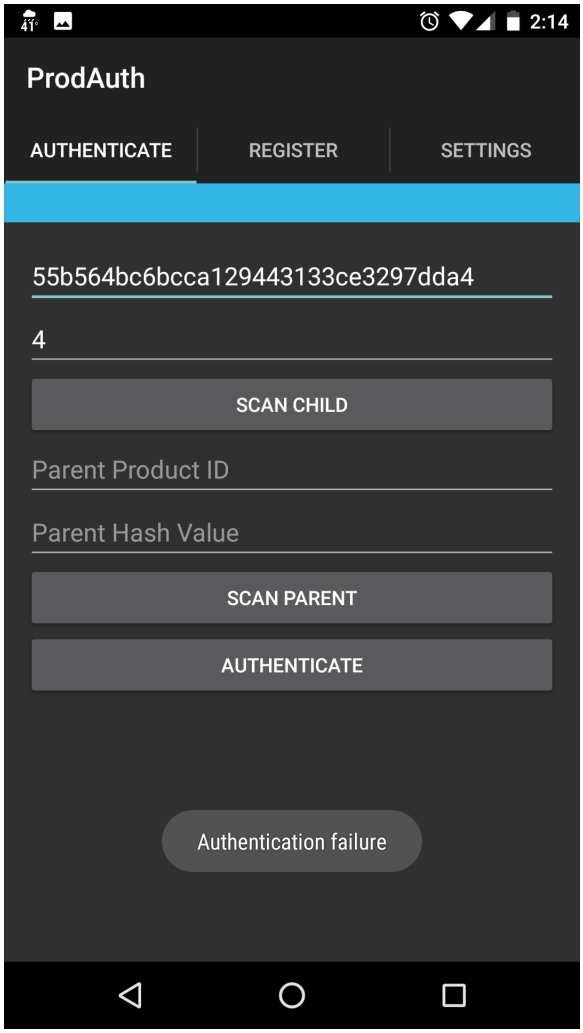

Figure 5.3: Failed Authentication

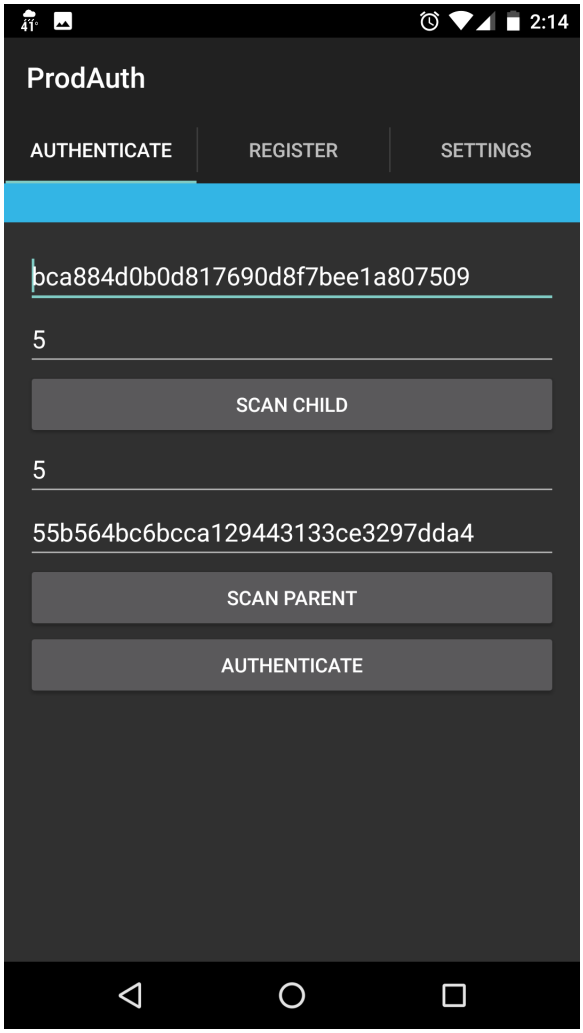

Figure 5.4: Combined Authentication

Chapter 6

Conclusion

Most research in product authentication use RFID tags, which can be expensive, complicated and need special hardware for use. We approached this problem using a different perspective and attempted to use user-friendly mechanisms for product authentication. Due to the ubiquity of smartphones with cameras, and the ease of use of being able to scan a simple QR code, product authentication can be made easier and more available to a general audience. Overall, we established that despite using a simpler medium like a QR code for storing security information, a protocol can be designed which can securely identify counterfeits and mitigate losses due to counterfeiting while empowering users to be more aware of the product they are using. It is fairly simple to build an application that piggybacks on this authentication scheme and allow companies to provide more detailed information about their product to users and engage with them.

A combined approach of using RFID tags along with printed QR Codes allows the whole supply chain to be securely authenticated and provides additional security features such as tracking, detection of loss, theft prevention, etc. Furthermore, this allows more secure but complex and expensive authentication techniques to be used at the supply-chain side where the cost is bearable by the manufacturer or distributor.

The consumers still get the benefits of having a low-cost solution for authenticating products conveniently.

This scheme can also be structured to work with barcodes or simple black printing of the ID and the hash value. A main assumption that we held while developing this scheme was that a counterfeiter could access the information in the QR Code. In other words, the information printed directly on a product was not assumed secure. Overall, our authentication scheme is a new twist on the research that has been done in this area. Most researchers concern themselves with the RFID tag, but that is too expensive to be considered for many small products. For example, you cannot put an RFID tag on every aspirin pill created. That is what this scheme looks at. We took the strategies implemented in RFID security and adapted them to create our scheme without an RFID tag.

In the future, this protocol can have a broader impact by using covert inks and specialized readers for an added layer of security at the cost of reduced availability to the general public [17]. Such a technique can be integrated into a product's supply chain to improve tracking by manufacturers and prevent loss, leakage or theft at any point. This scheme will have to be extended further to ensure that it can securely handle the complexities of a realistic supply chain while still maintaining simplicity and security.

Because of the widespread use of smartphones, other capabilities of these devices can be leveraged further in tackling counterfeits. Using the smartphone's location, an identified counterfeit can be geotagged and hotspots of counterfeiting can be recorded. This kind of data collected can be useful for investigative purposes for finding and tracking counterfeits.

Further work can also be done using similar schemes to authenticate digital media with embedded or steganographically hidden metadata since such an approach will rely on using static codes too.

The combined approach for authentication has multiple applications in securing various types of supply chains. An experimental approach of testing various methods of combined authentication will be useful in evaluating the best algorithms for the task for given authentication scenarios and use-cases. This combined approach can be expanded further to use multiple algorithms as well as a more complex graph of relationships between tags for complex real-world scenarios for authentication.

Bibliography

[1] Information technology – automatic identification and data capture techniques – qr code bar code symbology specification. ISO/IEC 18004:2015, International Organization for Standardization, Geneva, Switzerland.

[2] Information technology – radio frequency identification for item management – part 6: Parameters for air interface communications at 860 mhz to 960 mhz. ISO/IEC 18000-6:2010, International Organization for Standardization, Geneva, Switzerland.

[3] International Trademark Association. Fact sheets - counterfeiting, April 2015. `http://www.inta.org/TrademarkBasics/FactSheets/Pages/Counterfeiting.aspx`.

[4] Erwin A Blackstone, Joseph P Fuhr Jr, and Steve Pociask. The health and economic effects of counterfeit drugs. *American health & drug benefits*, 7(4), 2014.

[5] Jung-Sik Cho, Sang-Soo Yeo, and Sung Kwon Kim. Securing against brute-force attack: A hash-based rfid mutual authentication protocol using a secret value. *Computer Communications*, 34(3):391–397, 2011.

[6] Anil K Deisingh. Pharmaceutical counterfeiting. *Analyst*, 130(3):271–279, 2005.

[7] Frontier Economics. The impact of counterfeiting on governments and consumers. *A Report Commissioned by BASCAP (Business Action to Stop Counterfeiting and Piracy), An ICC Initiative*, 2009.

[8] Charlotte Eliasson and Pavel Matousek. Noninvasive authentication of pharmaceutical products through packaging using spatially offset raman spectroscopy. *Analytical Chemistry*, 79(4):1696–1701, 2007.

[9] Food, Drug Administration, et al. Combating counterfeit drugs. *US Department of Health and Human Services, Rockville, Maryland*, 2004.

[10] Dirk Henrici and Philipp Muller. Providing security and privacy in rfid systems using triggered hash chains. In *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*, pages 50–59. IEEE, 2008.

[11] Heiko Knospe and Hartmut Pohl. Rfid security. *Information security technical report*, 9(4):39–50, 2004.

[12] SK Kwok, Jacky SL Ting, Albert HC Tsang, WB Lee, and Benny CF Cheung. Design and development of a mobile epc-rfid-based self-validation system (mess) for product authentication. *Computers in Industry*, 61(7):624–635, 2010.

[13] Leslie Lamport. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–772, 1981.

[14] Mikko Lehtonen, Thorsten Staake, and Florian Michahelles. From identification to authentication–a review of rfid product authentication techniques. In *Networked RFID Systems and Lightweight Cryptography*, pages 169–187. Springer, 2008.

[15] Mikko O Lehtonen, Florian Michahelles, and Elgar Fleisch. Trust and security in rfid-based product authentication systems. *Systems Journal, IEEE*, 1(2):129–144, 2007.

[16] Ling Li. Technology designed to combat fakes in the global supply chain. *Business Horizons*, 56(2):167–177, 2013.

[17] Brian A Logue, Jamie Kern, Shelby Altena, Jacob Petersen, Sierra Rassmusan, Robert Oda, and Jon J Kellar. Countering counterfeiting of drugs: Unique fluorescent inks for direct printing onto pharmaceuticals. In *NIP & Digital Fabrication Conference*, volume 2015, pages 371–374. Society for Imaging Science and Technology, 2015.

[18] Miyako Ohkubo, Koutarou Suzuki, Shingo Kinoshita, et al. Cryptographic approach to privacy-friendly tags. In *RFID privacy workshop*, volume 82. Cambridge, USA, 2003.

[19] Bart Preneel. Cryptographic hash functions. *European Transactions on Telecommunications*, 5(4):431–448, 1994.

[20] Boyeon Song and Chris J Mitchell. Rfid authentication protocol for low-cost tags. In *Proceedings of the first ACM conference on Wireless network security*, pages 140–147. ACM, 2008.

[21] Irfan Syamsuddin, Tharam Dillon, Elizabeth Chang, and Song Han. A survey of rfid authentication protocols based on hash-chain method. In *Convergence and Hybrid Information Technology, 2008. ICCIT'08. Third International Conference on*, volume 2, pages 559–564. IEEE, 2008.

[22] Gene Tsudik. Ya-trap: Yet another trivial rfid authentication protocol. In *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, pages 4–pp. IEEE, 2006.

[23] Stephen A Weis, Sanjay E Sarma, Ronald L Rivest, and Daniel W Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *Security in pervasive computing*, pages 201–212. Springer, 2004.

[24] Liu Yang, Peng Yu, Wang Bailing, Qu Yun, Bai Xuefeng, Yuan Xinling, et al. Hash-based rfid mutual authentication protocol. *International Journal of Security & Its Applications*, 7(3):183–194, 2013.