Hash-Chain Based Product Authentication Scheme

Montana Earle (SUNY Potsdam)

Program Advisors
Dr. Manki Min; Research Advisor Dr. Brian Logue; REU Site Director Dr. Alfred Boysen; Professor Assisted by Harshith Keni, Grad Student, SDSU

Introduction

Counterfeiting has become a huge problem in the global market today and continues to grow. It is very common for a person to own a 'knock-off' of a product simply because it is cheaper. Most people think, what is the harm in a fake pair of sunglasses? The truth is that the funding from fake products funds the same trade routes where more dangerous counterfeits go through. For example, counterfeit medicine is a huge problem in third world country. Instead of solving the illness, these medicines can kill. Also, counterfeit computer chips can find their ways into defense systems.
Product authentication has become a very important part of the efforts to counteract counterfeiting efforts. There have been many different schemes developed to do this. Invisible inks and QR codes have become very popular, as well as RFID tags. But, one thing that needs work is the secure way of checking these tags and codes are what they are supposed to be. It is not enough to just put an invisible label on a product. The invisible label needs to be verified. This is where the networking side of product authentication comes in, as well as the scheme we are proposing here.

Terms to know:

Denial of Service Attack: This attack prevents communication, either by stopping all the messages to a certain target or sending so many that the target is overloaded and crashes.

Replay Attack: The attacker intercepts a message and replays it over and over again to try and make something happen that should not happen.

Insider Attacks: The attacker either has legitimate access to the network or is masquerading as someone who does. This type of attack sends messages from the inside and messes with the network that way.

Forward Security: If the attacker gets a message, they should not be able to construct the next one. They simply have what they got.

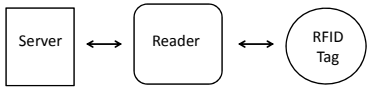
Mutual Authentication: Two parties in the communication authenticate that they are indeed talking to each other.

Other Schemes

Many different schemes have been developed to authenticate RFID tags. Our goal is to develop a protocol to authenticate static markings like a bar code or a QR code. We decided upon a QR code. But, to develop our scheme we had to look at what others had done. This involved looking at RFID authentication protocols.

RFID Authentication:

In it's most basic form, the RFID Tag's information is read by a Reader and verified with a Server.
The protocols are put in place to counteract security threats against this basic model.



We performed a comparison between the different protocols which is summarized in the following:

Table 1. Comparison of Protocols Dealing with attacks

Authentication Protocol	Forward Security	Anti- Denial of Service Attack	Anti-Replay Attack	Anti-Insider Attack	Mutual Authentication
Hash-based RFID Mutual Authentication Protocol [2]	Yes	No	Yes	No	Yes
YA-TRAP [3]	No	No	Yes	No	No
Triggered Hash Chain [4]	No	Yes	No	No	Yes
Secret Value Hash-based Mutual Authentication [5]	No	No	No	No	Yes
Cryptographic Approach to "Privacy-Friendly" Tags.[6]	No	No	No	No	No
RFID Authentication for Low-Cost Tags [7]	Yes	No	Yes	Yes	Yes

As you can clearly see, not a single one can prevent against all of the common attacks while also having the common secure characteristics. All of these RFID protocols except [7] use hash functions. There was a focus on researching these because that is a very integral part of the protocol that will be proposed later on in this paper. Also, some of these protocols were very focused on one aspect of security, while completely ignoring others. For example, [5] was concerned exclusively with mutual authentication, while [3] only worked on defending against replay attacks. Also, [6] focused on a good way to update the shared secret and completely ignored all security aspects. In our scheme we want to focus on all of these aspects and see what we can do to create a protocol that is secure from almost any attack.

Table 2. Time and Space Complexity of Protocols

Authentication protocol`	Time complexity	Space complexity
Hash-based RFID Mutual Authentication Protocol [2]	Tag: 4H + 1R Reader: 5H + 2R + 1K DB: 3H, 1K	Tag: 1L Reader: 1L DB: 2NL
YA-TRAP [3]	Tag: 1H + 1S + 2C Reader: DB: Comparisons (however many Ls in that hashtable)	Database: N(however many L's in that hashtable) Reader: clock Tag: 3L
Triggered Hash Chain [4]	Tag: 3H Reader: DB:3H	Tag: 1L Reader: DB: 5NL
Secret Value Hash-based Mutual Authentication [5]	Tag: 2H + 2M+ 1R Reader: R DB: (worst-case): O(2N)	Tag: 2L Reader: DB: 3NL
Cryptographic Approach to "Privacy-Friendly" Tags. [6]	Tag:2H Reader: DB:2NiH(worst-case)	Tag: 1L Reader: DB: 2NL
RFID Authentication for Low-Cost Tags [7]	Tag: 3H Reader: 1R DB: 2NH, 1H	Tag: 1L Reader: 1L DB: 4NL

Key:
H = 1 hash computation
S= subtraction computation
R = one random number computation
i = round number
K= one symmetric encryption/decryption
L = 1 variable stored (128 bit)
N = number of tags stored in database
M= mod operation
X = XOR operation

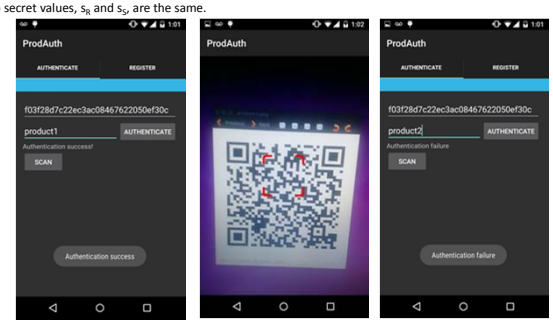
Yet, a drawback to this is you have to be very careful on how complex you make a protocol. It can be the most secure protocol ever, but it may not be very practical if it is too complex and/or takes too much time. The above table is the complexities found in the RFID protocols. There are always tradeoffs between the level of security and the complexity of the scheme.

Drawbacks of RFID

There are many drawbacks to the RFID system. First off, there is a cost to making an RFID tag. It is not very feasible to put one on every individual item you want to authenticate. The expense would not be cost effective. Also, making an RFID tag tamper resistant is highly expensive and impractical. So, there is really no way to feasibly prevent tampering. Additionally, an RFID tag can only hold a limited amount of data. Not to mention, the wireless communication between a RFID tag and a reader is vulnerable to eavesdropping.
To counteract these drawbacks, our proposed scheme will use a QR code instead. The cost of printing a QR code onto each individual product is much less than creating an RFID tag for each one. Also, if the information is changed, the code would simply be declared unauthentic. Additionally, a QR code can hold and transmit much more data than a RFID tag. Not to mention, if someone reads the QR code, then they do not have any information that is not open to the public.

Our Proposed Authentication Protocol

At the beginning, each element of the scheme holds certain elements:
Server:
--Table of reader ID_R's with s_S for each one
--The number of times each product can be authenticated, N
--Hardcoded secret v
--Random variable k
--A table in which for each registered product ID_P the first hash value (k_1), and for each reader ID_R how many times a product has been authenticated (i) is stored.
Reader :
--Hardcoded secret v
--Random variable k
--Secret Value, s_R
QR Code:
--Product ID_P
--Last Hash Value, $hash_n$
The two secret values, s_R and s_S, are the same.



Server	Reader	QR Code
		ID || $hash_n$
	Calculates $g(g(s_R))$ and k = $g(s_R)$	
	$g(g(s_R))$ || ID_P || ID_R	
Calculates $g(g(s_S))$ and k = $g(s_S)$ If $g(g(s_R))$ = $g(g(s_S))$, the reader is authenticated If ID_R and ID_P are in the table, looks up k_i and i If i > N, authentication fails, MSG := error Else Calculates MSG := $h^{N+1}(k_1)$ Else authentication fails, MSG := error s_S = $f(s_S$ XOR v) MSG || i ||k ➡ Else the reader is not authentic and authentication fails		
	If k = $g(s_S)$, the server is authenticated and s_R = $f(s_R$ XOR v) Calculate $hash_n$' based on $h^{N+1}(k_1)$ If $hash_n$ = $hash_n$', the product is Authenticated, MSG := valid Else, MSG := error ⬅ MSG || $g(g(s_R))$ || ID_R || ID_P s_R = $f(s_R$ XOR v) Else server is not authenticated	
Calculates $g(g(s_S))$ If $g(g(s_R))$ = $g(g(s_S))$, the reader is authenticated If you get a valid message, i++ s_S = $f(s_S$ XOR v) Else the reader is not authenticated		

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.
Through the use of shared secrets that update, the server and the reader mutually authenticate each other. Also, the update process of the secret, s, and the use of the hash chain to check $hash_n$ 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 used over and over again. Also, this process of authentication prevents a denial of service attack. Since the message is authenticated first, the server or reader will only do a little work and then ditch the message.
Also, a concern we came up with is that a fake reader could be put up on the app market. We also figure that if someone is going to go through the trouble of authenticating products, they are going to do their research on what app to use. As long as companies state which app is the real app on their websites, people will know which app they should use. Another defense against a fake app is that if a fake reader is registered to a server, it will be useless after one communication. It will not have the hardcoded secret and thus cannot update the shared secret correctly.
Also, the limit N was set so that a product can only be authenticated so many times. This helps prevent against attackers taking a valid QR code and copying it onto many counterfeit products. This way there is a limit to these products being authenticated. Then, all of them will be declared unauthentic.

Authentication Protocol	Forward Security	Anti- Denial of Service Attack	Anti- Replay Attack	Anti- Insider Attack	Mutual Authentication
Our protocol	Yes	Yes	Yes	Yes	Yes

Complexity

Our protocol is a little more complex compared to other RFI-based product authentication schemes. However considering the relatively small N, the complexity can be comparable to other RFID-based schemes. If the server precomputes the hash chain and store them, we can sacrifice the space complexity in favor of the time complexity.

Authentication protocol	Time complexity	Space complexity
Our protocol	Server: (5+N-i)H + 2X Reader: (6+i)H + 2X	Server: (R+3+2P+PR)L Reader: 3L

Key:
H = 1 hash computation
L = 1 variable stored (128 bit)
N = number of times a tag can be authenticated
X = XOR operation
P = number of Products
i = round number
R= number of Readers

Screenshots

A QR Code is read in through an app. This app/Reader then transfers the ID_P and $hash_n$ to the server and responds with whether or not this is authentic. That is what a user sees. Behind the scenes, the app/Reader and the Server are using our authentication protocol.

References

1. Stallings, William. *Cryptography and Network Security.* Sixth ed. Pearson Education, 2014. Print.
2. Yang, Liu , Peng Yu, Wang Bailing, Qu Yun, Bai Xuefeng, Yuan Xinling and Yin Zelong. "Hash-based RFID Mutual Authentication Protocol." *International Journal of Security and Its Applications.* 7.3 (2013): 183-194. Web.
3. Tsudik, Gene. "YA-TRAP: Yet Another Trivial RFID Authentication Protocol." *Pervasive Computing and Communications Workshops.* (2006): Web.
4. Henrici, Dirk, and Philipp Muller. "Providing security and privacy in RFID systems using triggered hash chains." *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on.* IEEE, 2008.
5. Cho, Jung-Sik, 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 (2011): 391-397.
6. Ohkubo, Miyako, Koutarou Suzuki and Shingo Kinoshita. "Cryptographic Approach to "Privacy-Friendly" Tags."
7. Song, Boyeon, and Chris J. Mitchell. "RFID authentication protocol for low-cost tags." *Proceedings of the first ACM conference on Wireless network security.* ACM, 2008.

Acknowledgements

This work was made possible by the National Science Foundation REU Security Printing and Anti-Counterfeiting Site EEC-1263343. Thanks to advisor Dr. Manki Min and REU site director Dr. Brian Logue for their direction and guidance, Dr. Alfred Boysen for his English help and support, as well as Harshith Keni for his help in this research.