The Global Language of Business

EPC Write-Protection Recommendation

Document Purpose

The purpose of this document is to outline a common process for write-protecting the EPC memory bank of UHF RFID tags while avoiding technical details that may become antiquated or vary between RFID chip supplier. While some RFID applications (such as those in a closed-loop environment) may not opt to write-protect RFID tags, many RFID solutions rely on tag data to be write protected. The EPC memory bank (MB01) contains serialized identification keys (EPC) that allow the unique identification of items to which the RFID tag is attached to. If the desire is to possibly change the EPC tag data after it is encoded, then the approach of "Locking" the tag (and read-protecting the password) may be used. If the desire is to not allow *any* changes to the EPC-enabled RFID tags, then Permalocking the EPC memory bank may be used. This document is intended for use by RFID solution providers, developers and organizations that provide EPC-encoded RFID tags at source. It includes detailed process examples of a technical nature. The expectation is that tag auditing agencies will apply the below-outlined process if asked to validate tags on behalf of a retailer that is opting to (reversibly) lock or permalock tags.

Overview of Write-protecting RFID Tags

An RFID chip implements a set of LOCK Action bits for every memory bank. Each memory bank's Lock status is described by two bits. One for the pwd-write (or pwd-read/write) and a second one for permalock.

Let's take the example of the EPC memory bank (MB01). If locked (by the use of a non-zero access password) or if permalocked, you can read it but you cannot write it. Depending on the value of these two bits you are allowed to:

	Pwd-write=0	**Pwd-write=1**
Permalock=0	You can read and write the EPC memory and you can change the protection status. This is not recommended.	You can read the EPC memory but in order to write it you must know the access password[1]. This means that the EPC memory is password write protected[2]. You can also change the protection status. This is the preferred way to reversibly lock the EPC memory.
Permalock=1	Under this condition the EPC memory will be always writeable (and readable) and this status cannot ever be changed. There is no way to write-protect the EPC memory.	You can read the EPC memory, but you can never write it whatever the state. You will never be able to change the protection status. This is the preferred way to permalock the EPC memory.

There are two options for locking: permanent locking or reversible locking by knowing the access password. For the reversible lock[3] of the EPC memory, it is important to have an access password (non-zero) programmed and not readable (either locked or permalocked). To get information on the status of a memory location it is possible to follow one of the two methods detailed later in the document under the title of "Auditor Verification Process for EPC Memory Locking". One can either use the mandatory method available on all EPC-compliant tags to make read/write trials to memory and see the response of the tag, or, secondly, leverage optional features to read a bit exposing the lock status.

---

[1] This only applies to RFID chips that support the optional *Access* command (ie. RFID chips for which the access password can be set to a non-zero value)

[2] Note that the access password can be pwd-read/write locked such that the access password is needed to read the password. Leaving the access password unlocked (pwd-read/write = 0) means the password will be readable by anyone. In the case of it being readable, this is as secure as writing your laptop password on a post-it attached to it.

[3] If you want to make use of a reversible lock, make sure that the *Access* command is supported by the RFID chip

Vendor Program Encoding Recommendation:

Identifiers encoded in the EPC memory of tags should either be reversibly locked or permalocked.

The encoding recommendation for programs where EPC memory bank values should be reversibly locked follows[4]:

EPC Memory Reversible Locking Steps: (Pwd-write=1 and Permalock=0)

1. Encode (write) the EPC memory with the desired value (SGTIN-96, GIAI-96, etc.) as per the GS1 Tag Data Standard.
2. Read the EPC memory to verify that the written data is correct.
3. If the written data is correct, go to step 5)
4. If the written data is not correct, write the EPC memory again (you can increase the write power and/or verify that the tag to be encoded is in the reader's field of view and/or verify that there are no other tags in the reader's field of view) and go to step 2.
5. Write the access password to a non-zero value.[5]
6. Read the access password to verify that the value is correct.
7. If the value is correct, go to step 9.
8. If the value is not correct, write the access password again (you can increase the write power and/or verify that the tag to be encoded is in the reader's field of view and/or verify that there are no other tags in the reader's field of view) and go to step 6.
9. Using the correct access password and the *Lock* command, lock the access password and EPC memory so that the access password will be pwd-read/write protected and the EPC will be pwd-write protected.[6]
10. Verify that access password is locked by trying to read it from the **open** state.[7]
11. If the access password is not readable, go to step 13.
12. If the access password is still readable, go to step 9.
13. Verify that EPC memory is locked by trying to change its value from the **open** state.
14. From the **open** state (just after the tag has been inventoried), try to rewrite its EPC memory.
15. If the value of the EPC memory can be changed, go to step 1.
16. If the value of the EPC memory cannot be changed, process is completed.

The encoding recommendation for programs where EPC memory bank values should be permalocked follows:

EPC Memory Permalocking Steps: (Pwd-write=1 and Permalock=1)

1. Encode (write) the EPC memory with the desired value (SGTIN-96, GIAI-96, etc.) as per the GS1 Tag Data Standard.
2. Read the EPC memory to verify that the written data is correct.
3. If the written data is correct, go to step 5.
4. If the written data is not correct, write the EPC memory again (you can increase the write power and/or verify that the tag to be encoded is in the reader's field of view and/or verify that there are no other tags in the reader's field of view), go to step 2.
5. Permalock the EPC memory by issuing a *Lock* command. This will write-protect the EPC value which can never again be changed. (Some chips may require that a password be set first or the issuing of a lock command to lock-permalock all memory as outlined in the Gen2 specification.)
6. Verify that EPC memory is permalocked by trying to change its value from the **secured** state.
7. If the value of the EPC memory can be changed go to step 1.
8. If the value of the EPC memory cannot be changed, process is completed.

---

[4] This only applies to RFID chips that support the *Access* command ie. RFID chips for which the access password can be set to a non-zero value

[5] This value has to be kept secret and it is recommended that each single tag has a different access password. For that purpose, some algorithms provide ways to derive unique access passwords from the unique value of the TID.

[6] Permalocking the access password is also an option but this will make the password permanently read protected whatever the state and its value can never again be changed.

[7] From the Gen2 spec, after tags are inventoried, they will transition to the **open** state if the tag has a non-zero access password. Writing a locked EPC, reading a locked access password or locking the tag with the Lock command will fail from the **open** state – these operations require the tag to be in the **secured** state. A tag will transition from the **open** state to the **secured** state by the reader issuing an *Access* command sequence using the correct access password.

Note: In order to permalock the EPC memory, you may not need to encode a non-zero access password (some chip suppliers may require a non-zero password). A tag that is properly permalocked will not allow the EPC memory to be written to even in a **secured** state (e.g., even when leveraging the zero or non-zero password to enter a **secured** state). Either leave the access password to zero (if advisable by the chip supplier) or ensure that the access password is not read-protected[8].

Auditor Verification Process for EPC Memory Locking

The following processes may be used to confirm reversible lock or permalock per the above encoding recommendation options. The first is supported by every EPC-compliant chip. The second process leverages optional features.

This process is supported by every EPC-compliant chip:

1. Just after the tag has been inventoried, try to rewrite the EPC memory
   - If the EPC value can be changed, the EPC is not reversibly locked nor permalocked: test failed
   - If the EPC value has not been changed, the EPC is write-protected: proceed to next step
2. Attempt to read the access password
   - If the password cannot be read: **test passed (tag is either permalocked or locked)**
   - If the password can be read, continue to the next step
3. Issue an *Access* command with the access password to enter the **secured** state.
4. Attempt to unlock the EPC memory and write a different EPC value
   - If the EPC can be unlocked and the value has been changed[9]: test failed
   - If the EPC has not been changed, the EPC is permalocked: **test passed**

This alternative process leverages optional features:

As an alternative, per the Gen2 specification, if a tag chip supports verifying EPC and access password lock status by reading lock status bits in tag memory, the following process may be followed.

1. Read the EPC lock status bit(s) to determine EPC lock status.
   - If the EPC status bits confirm the EPC memory is permalocked, the EPC memory is write-protected: test passed
   - If the EPC status bits confirm the EPC memory is not locked or permalocked, the EPC memory is not write-protected: **test failed**
   - If the EPC status bit(s) confirm the EPC memory is locked but not permalocked, read the access password lock status bits.
     - If the access password status bits confirm the access password is not locked or permalocked, the access password is readable so the EPC memory is not properly write-protected: **test failed**
     - If the access password status bits confirm the access password is locked or permalocked, the access password is not readable from the **open** state so the EPC memory is write-protected: test passed

Additional Technical Information

This document is intended as a brief overview of the suggested general approach to write-protect tags. For additional technical information that is specific to a particular chip, please contact the relevant chip manufacturer.

Links to standards:

- EPC UHF Air Interface Protocol: https://www.gs1.org/standards/epc-rfid/uhf-air-interface-protocol
- GS1 Tag Data Standard: https://www.gs1.org/standards/tds

[8] A permalocked tag EPC encoding cannot be changed even when using the access password. Allowing the access password to be read for permalocked tags enables an auditor to confirm that the memory is permalocked.
[9] This means that the EPC memory is locked with an access password which is not read-protected

Proprietary Statement
This document contains proprietary information of GS1 US. Such proprietary information may not be changed for use with any other parties for any other purpose without the expressed written permission of GS1 US.

Improvements
Improvements and changes are periodically made to publications by GS1 US. All material is subject to change without notice. Please refer to GS1 US website for the most current publication available.

Disclaimer
Except as may be otherwise indicated in specific documents within this publication, you are authorized to view documents within this publication, subject to the following:

1. You agree to retain all copyright and other proprietary notices on every copy you make.
2. Some documents may contain other proprietary notices and copyright information relating to that document. You agree that GS1 US has not conferred by implication, estoppels, or otherwise any license or right under any patent, trademark, or copyright (except as expressly provided above) of GS1 US or of any third party.

This publication is provided "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. Any GS1 US publication may include technical inaccuracies or typographical errors. GS1 US assumes no responsibility for and disclaims all liability for any errors or omissions in this publication or in other documents which are referred to within or linked to this publication. Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

Several products and company names mentioned herein may be trademarks and/or registered trademarks of their respective companies. GS1 US does not, by promulgating this document on behalf of the parties involved in the creation of this document, represent that any methods, products, and/or systems discussed or recommended in the document do not violate the intellectual property rights of any third party. GS1 US has not performed a search to determine what intellectual property may be infringed by an implementation of any strategies or suggestions included in this document. GS1 US hereby disclaims any liability for any party's infringement of intellectual property rights that arise as a result of any implementation of strategies or suggestions included in this document.

This publication may be distributed internationally and may contain references to GS1 US products, programs, and services that have not been announced in your country. These references do not imply that GS1 US intends to announce such products, programs, or services in your country.

GS1 US shall not be liable for any consequential, special, indirect, incidental, liquidated, exemplary, or punitive damages of any kind or nature whatsoever, or any lost income or profits, under any theory of liability, arising out of the use of this publication or any content herein, even if advised of the possibility of such loss or damage or if such loss or damage could have been reasonably foreseen.

GS1 US HEREBY DISCLAIMS, AND YOU HEREBY EXPRESSLY RELEASE GS1 US FROM, ANY AND ALL LIABILITY RELATING TO YOUR COMPLIANCE WITH REGULATORY STANDARDS AND LAWS, INCLUDING ALL RULES AND REGULATIONS PROMULGATED THEREUNDER. GS1 US MAKES NO WARRANTIES OF ANY KIND RELATING TO THE SUITABILITY OF THE GS1 STANDARDS AND THE SPECIFIC DOCUMENTS WITHIN THIS PUBLICATION TO COMPLY WITH ANY REGULATORY STANDARDS, LAWS, RULES AND REGULATIONS. ALL INFORMATION AND SERVICES ARE PROVIDED "AS IS."

*GS1 US employees are not representatives or agents of the U.S. FDA, and the content of this publication has not been reviewed, approved, or authorized by the U.S. FDA. The following information contained herein is for informational purposes only as a convenience, and is not legal advice or a substitute for legal counsel. GS1 US Inc. assumes no liability for the use or interpretation of the information contained herein.

No Liability for Consequential Damage
In no event shall GS1 US or anyone else involved in the creation, production, or delivery of the accompanying documentation be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other loss) arising out of the use of or the results of use of or inability to use such documentation, even if GS1 US has been advised of the possibility of such damages.

IAPMO
In this publication, the letters "U.P.C." are used solely as an abbreviation for the "Universal Product Code" which is a product identification system. They do not refer to the UPC, which is a federally registered certification mark of the International Association of Plumbing and Mechanical Officials (IAPMO) to certify compliance with a Uniform Plumbing Code as authorized by IAPMO.

*If applicable