SANDIA REPORT

SAND2015-10101
Unlimited Release
Printed November 2015

Authentication Without Secrets

Lyndon G. Pierson and Perry J. Robertson

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: http://www.ntis.gov/search

SAND2015-10101
Unlimited Release
Printed November 2015

Authentication Without Secrets

Lyndon G. Pierson (Ret.)
Networked Systems Surveillance & Assurance
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-1072

Perry J. Robertson
RF and Opto Microsystems
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-0341

Abstract

This work examines a new approach to authentication, which is the most fundamental security primitive that underpins all cyber security protections. Current Internet authentication techniques require the protection of one or more secret keys along with the integrity protection of the algorithms/computations designed to prove possession of the secret without actually revealing it. Protecting a secret requires physical barriers or encryption with yet another secret key. The reason to strive for "Authentication without Secret Keys" is that protecting secrets (even small ones only kept in a small corner of a component or device) is much harder than protecting the integrity of information that is not secret. Promising methods are examined for authentication of components, data, programs, network transactions, and/or individuals. The successful development of authentication without secret keys will enable far more tractable system security engineering for high exposure, high consequence systems by eliminating the need for brittle protection mechanisms to protect secret keys (such as are now protected in smart cards, etc.). This paper is a re-release of SAND2009-7032 with new figures numerous edits.

CONTENTS

Executive Summary 7

Introduction 9

Infosets, Secrets and Integrets 13
- Infoset 13
- Secrets 14
- Integrets 15
- Protection for an Integret 15
- Protection for a Secret 16
- "Authentication without Secrecy" is not the same as "Authentication without Secrets" 18
- Message Authentication 19
  - Authentication and Identity 20
  - Two Factor Authentication 22
  - Authentication and Authorization 23
- Quantum Authentication 24
- Digital Signatures 25
- Digital Signature Algorithm 26
- Communications Between Components 27
- Authenticating Components 28

Biometric Measurements and Physically Unclonable Functions 31
- Biometrics 31
- Physically Unclonable Functions (PUFs) 33

Metrics for Brittleness of Authentication Systems using Secret Keys 35

Conclusion 39

FIGURES

Figure 1. An Infoset and relationships to Author and Viewer. 14
Figure 2. Using the SHA with the DSA (from NIST). 27
Figure 3. Elements of a Typical Biometric Authentication System. 31
Figure 4. Remote Biometric Authentication. 32
Figure 5. Brittleness of Authentication due to Secret-Keeping. 36
Figure 6. Design Exposure Vector Example. 37
Figure 7. Operational Exposure Vector Example. 38

EXECUTIVE SUMMARY

This paper is a re-release of the original paper "Authentication Without Secrecy" SAND2009-7032. This paper has been re-released with new figures and numerous edits. The document has been reviewed for Unlimited Release.

Authentication of identity and of message origin is a key element of trust in any secure communications or information processing system.

Current Internet authentication techniques require the protection of one or more secret keys along with algorithms/computations designed to prove possession of the secret without actually revealing it (so that the authentication process can be exercised multiple times without generating and qualifying a new secret) [1].

The reason to strive for "Authentication without Secret Keys" is that protecting secrets (even small ones only kept in a small corner of a component or device) is much harder than protecting the integrity of information that is not secret (an integret). This is especially important for systems with high exposure to the adversary in which a sophisticated adversary has physical access to the "creation" portion of the life cycle, and at least logical access (perhaps through the internet) for the "deployment" portion of the life cycle. In these high exposure systems, sophisticated adversaries may have a high likelihood of being able to reverse engineer and circumvent even secret protection measures and then to extract the operational secret(s), thus enabling the spoofing of the authentication system. While the development of methods of authentication without secrets will benefit the security of high exposure systems, the security of lower exposure systems (in which secret protection measures to prevent secret extraction are more effective) will also benefit from these techniques.

Protecting a secret requires physical barriers or encryption with yet another secret key. If the protections for this secret are breached, the availability of the secret to the adversary can result in a catastrophic failure of the authentication mechanism, since the adversary then has sufficient information to spoof the authentication. This tendency for an authentication system to fail catastrophically if the operational secret is revealed is referred to as "brittleness" elsewhere in this report. Protecting an Integret[2] requires comparison with multiple copies held in a manner that the majority of copies are difficult to subvert, and/or transparent[3] mechanisms to assure against unauthorized modification of each individual copy of the data. For authentication systems that can be carefully designed to depend on Integrets rather than Secrets, if the Integret becomes known by the adversary, there is no catastrophic failure of the authentication system[4], and in this sense the system is more resilient and less brittle.

---

[1] For example, one of the most advanced authentication systems in use today is a challenge-response smart card authentication system in which a the smart card contains a secret key and is able to use it to decrypt and respond to a specially encrypted challenge. These kinds of systems depend on assurance of "unique possession" of the secret key, else some other party would also be able to successfully respond to the encrypted challenge, thus spoofing the authentication system.

[2] For the purposes of this paper, an *integret* is defined as data intended to be reliably known by many entities (in contrast to a secret which is data intended to be reliably known by very few well specified entities).

[3] Here we mean transparency in the "treaty verification" sense that multiple parties are assured "visibility" into the existence and proper operation of the protection mechanisms.

All electronic message authentication schemes in use today require protection of a small secret against extraction by an adversary over some portion of the life cycle[5]. This LDRD effort is to examine feasibility of refining an authentication approach that would enable measurable gains in cyber security by eliminating the need to protect a small secret from extraction by an adversary, while accomplishing robust authentication of Internet transactions. The possible alternatives (to secret-keeping authentication systems) are to be examined for suitability for authentication of components, data, programs, network transactions, and/or individuals)

What if it were possible for two components to communicate and authenticate each other without the use of a secret (key)? There are many advantages to this type of technique including reduced manufacturing cost and no risk to the exposure of the secret key (because there is none!). The successful development of authentication without secret keys will enable far more tractable system security engineering for high exposure, high consequence systems by eliminating the need for brittle protection mechanisms to protect secret keys (such as are now protected in smart cards, etc.).

This work has focused on 1) authentication systems that measure "hard to clone" characteristics (using concepts patterned after biometric techniques of authenticating individual human characteristics) and 2) authentication systems in which the secret keys, while not eliminated, are designed to reside in parts of the life cycle or environments that are more easily protected against extraction, and 3) system metrics that help understand and compare the fragility of these systems due to the catastrophic failure associated with loss of authentication secrets.

This report concludes that elimination of "secret-keeping" from our authentication systems is possible for certain applications, but known techniques involve tradeoffs between computational complexity of secret-keeping techniques and the large volume of data processing required for techniques that measure unique (hard to clone) non-secret characteristics. In the absence of a proof that low-overhead non-secret authentication techniques do not exist, further research in this area could result in high payoff. An initial, coarse system metric was formulated for comparing the brittleness/fragility of authentication systems that depend on secret-keeping. Refinement and augmentation of this metric and/or similar metrics will enable careful analysis and improvement of the resilience of authentication systems, especially in high-exposure environments.

---

[4] The validation of a digital signature, for example, requires processing with a well-known public key to validate the signature and thereby the origin of the data. The creation of the signature required processing with a secret, but for the purpose of this example, we presume that the signature process is well protected in a low-exposure environment with guards and guns. We say therefore, that the validation phase of a public key digital signature method is far less brittle than the signature phase.

[5] In contrast with certain human-culture based authentication mechanisms such as human recognition of the voice used to deliver a verbal message, for example.

INTRODUCTION

In this paper, we seek to describe "Authentication without Secret Keys", which is even more difficult than "authentication without secrecy" (of messages) that has been studied in the literature since Gus Simmons landmark efforts[6]. The reason to strive for "Authentication without Secret Keys" is that protecting secrets (even small ones only kept in a small corner of a component or device) is harder than protecting the integrity of information that is not secret. This is especially important for systems with high exposure to the adversary in which a sophisticated adversary has physical access to the "creation" portion of the life cycle, and at least logical access (perhaps through the internet) for the "deployment" portion of the life cycle. In these high exposure systems, sophisticated adversaries may have a high likelihood of being able to reverse engineer and circumvent even secret protection measures and then to extract the operational secret(s). While the development of methods of authentication without secrets will benefit the security of high exposure systems, the security of lower exposure systems (in which secret protection measures to prevent secret extraction are more effective) will also benefit from these techniques.

Authentication and encryption are related *but different* information technologies. The concept of authentication relates to either insuring the identity of the person to whom you are communicating or insuring that the message has not been modified. Encryption is generally used to insure message confidentiality, that is you are trying to keep the contents of a message out of the hands of those other than the intended receiver. Encryption can also be used in some authentication systems in order to protect a secret (private) key from disclosure. We take a more detailed look at the relationship between encryption and authentication in the next section.

Today, secure message transmission is such a common part of our daily lives that we sometimes have difficulty separating the concepts of confidentiality and authenticity. It has become common practice to cryptographically protect commerce and national secrets using encryption techniques, both symmetric and asymmetric. Symmetric encryption utilizes easily related (if not identical) cryptographic keys to both encrypt and decrypt a message. Asymmetric encryption utilizes a unique set of keys where not only are the encryption and decryption keys different, but they are related by a "computationally complex" problem that makes it nearly impossible to derive one key by having knowledge of the other[7]. Often called a public key cryptosystem, since a set of public keys are held in a public directory, this system was first described by Diffie and Hellman in 1976[8][9]. Both symmetric encryption (such as NIST's DES and AES) and public key encryption (such as RSA) have been used in various ways to protect secrets used for authentication. The most obvious method of employing symmetric encryption is to protect transmission of a password, but if known by both parties there is no method of non-repudiation

---

[6] G. J. Simmons, "Message authentication without secrecy," in AAAS Selected Symposia Series (G. Simmon, e d .) , pp. 105-139, 1982. (In the context of Simmon's work, "without secrecy" has come to mean "while sending the message in the clear", even though these techniques still require secret cryptovariables.)

[7] Ed. G. J. Simmons, "Secure Communications and Asymmetric Cryptosystems," AAAS Selected Symposium Series 69, from the 1980 AAAS National Annual Meeting in San Francisco, CA, Westview Press, 1982.

[8] Public key cryptography was independently discovered by Merkle this same year.

[9] It is interesting to note that while the "Diffie-Hellman" key exchange can be used to key up a secure communication channel, but by itself does nothing to assure knowledge of the identity of the communicating parties to each other For this reason it is often called "unauthenticated Diffie-Hellman" key exchange.

between the parties (either party might have given the password). A variation on using symmetric encryption for authentication is called "one-way encryption" in which the password is encrypted/decrypted in the transmission channel (to keep from prying eyes), then encrypted (but not decrypted) before comparing with a database of "ciphertext passwords". In this way, even if the password database were compromised, it would still be hard for an adversary to recover the original passwords (since only the ciphertext of the passwords is stored), and the authentication secret (password) is protected through a major part of the system. Another preferred method uses a challenge-response technique with asymmetric (public key) cryptography in which a random number is encrypted with a person's public key (challenge) and sent to the holder of the private key who decrypts the number with his/her private key and sends it back (response), proving that the interaction is with the holder of the private key (note the importance of "unique possession" of the private key).

In any case, all these systems rely upon a secret, and that secret requires protection from disclosure. This secret is used to encrypt a communication and keep it from being disclosed, or to prevent the spoofing of the authentication by anyone who does not have the secret. Public key encryption actually generates two secrets for each security association (one for each end of a bidirectional communication session), doubling the problem of protection of the secrets[10]. However, in the case of authentication, it may be possible to provide strong authentication (I am who I am) between two entities without the need for a secret.

Simmons (1982) described a thought experiment where two accomplices in a crime were arrested and locked in separate cells. The two prisoners are permitted to communicate via letters passed to the warden. That gives the warden a chance to view the messages. The prisoners need to know if the letter they receive from the other prisoner is really from the other prisoner or from the warden pretending to be the other prisoner. The two prisoners establish a "subliminal channel[11]" in full view of the warden. The subliminal channel is used to pass information that provides authentication of the sender. The messages themselves contain no secret (with respect to the warden) information. **Simmons asserted that authentication can be accomplished by introducing prearranged redundant information into the message.** The presence of this additional information indicates that the message is genuine. (The pre-arrangement of this redundancy amounts to a shared secret, of course.) This authentication information is cryptographically bound to the message to prevent the warden from stripping it off and using it to authenticate some other message[12]. From Simmons:

> The essential points to an authentication without secrecy channel are that;
>
> a) the receiver authenticates a message through the presence of Hr bits of redundant – expected – information in the decrypted cipher,

[10] Symmetric encryption also requires keeping a secret on each end of a communication, but it is the same secret shared by the two ends.

[11] A subliminal channel is a covert channel within a known, authorized channel.

[12] Thus, Simmons' authentication with out secrecy means authentication in which the message itself is not secret, even though the redundant information forming a "message authentication check" or the means to generate it is kept secret. (So in this context, "authentication without secrecy" still requires a small secret.)

b) the host to the communication channel verifies that nothing has been concealed by decrypting the ciphers and verifying that the resulting message is precisely what he expected based on a foreknowledge of the message.

As mentioned before, the system is operationally different for the host depending on whether the cryptoalgorithm is single or two key, since this determines whether he can check for concealed information before or after the exchange occurs. However, this does not alter the way in which he satisfies himself that nothing is concealed – namely, that the cipher decrypts to an expected message.

INFOSETS, SECRETS AND INTEGRETS

Infoset

In the context of this document, an infoset (short for "information set") is a well described or delineated set of "bits" that is created by an author or defined by a relationship specified by an author[13]. The relationship is diagramed in Figure 1. An author is a person or persons acting together, or a device acting on behalf of a person or persons acting together, that originates or gives existence to the information set. Examples of infosets include the "as-built" design of a complex system, a username, a password, a public key, a private key, etc. For example, a full or partial specification of an Integrated Circuit is also an infoset.

The author of an infoset is not presumed to have perfect memory and therefore may need to refer to a previously recorded version of the infoset. An infoset has integrity (high, medium, low) if its information continues to be known with high (or medium or low) certainty to its author and to other parties who access the information. If the infoset is restricted from being viewed by other parties, then it also has some secrecy (high, medium, low), depending on the extent to which the confidentiality is shared (see definition of secrecy below).

So an infoset has an author or originator who can't always remember the data contents (and therefore may need to be a viewer also), and other entities (viewers and non-viewers) who have high or low certainty regarding the data set, and maybe a manager who designates who should have high certainty regarding this infoset and who should have high uncertainty regarding this infoset. Typically the author/originator always has permission to view the infoset.

We further define an Infoset to consist of data and meta-data (information about the data), and possibly some redundant data used to validate the integrity of the data. The first element of the meta-data pertains to the existence of the infoset. Meta-data can be nested (infosets contained within infosets) and the boundary between data and meta-data could theoretically move back and forth (data bits can become meta-data and vice-versa, especially with regard to secrecy and integrity issues described below. Moving data bits to meta-data for an infoset for which the data is secret can result in improper disclosure (or increased sensitivity to disclosure) by "data aggregation", or gathering too much meta-data about the secret data.)

[13] Not to be confused with the XML use of the term "infoset" which usually involves "joins" of specific data sources or structures.

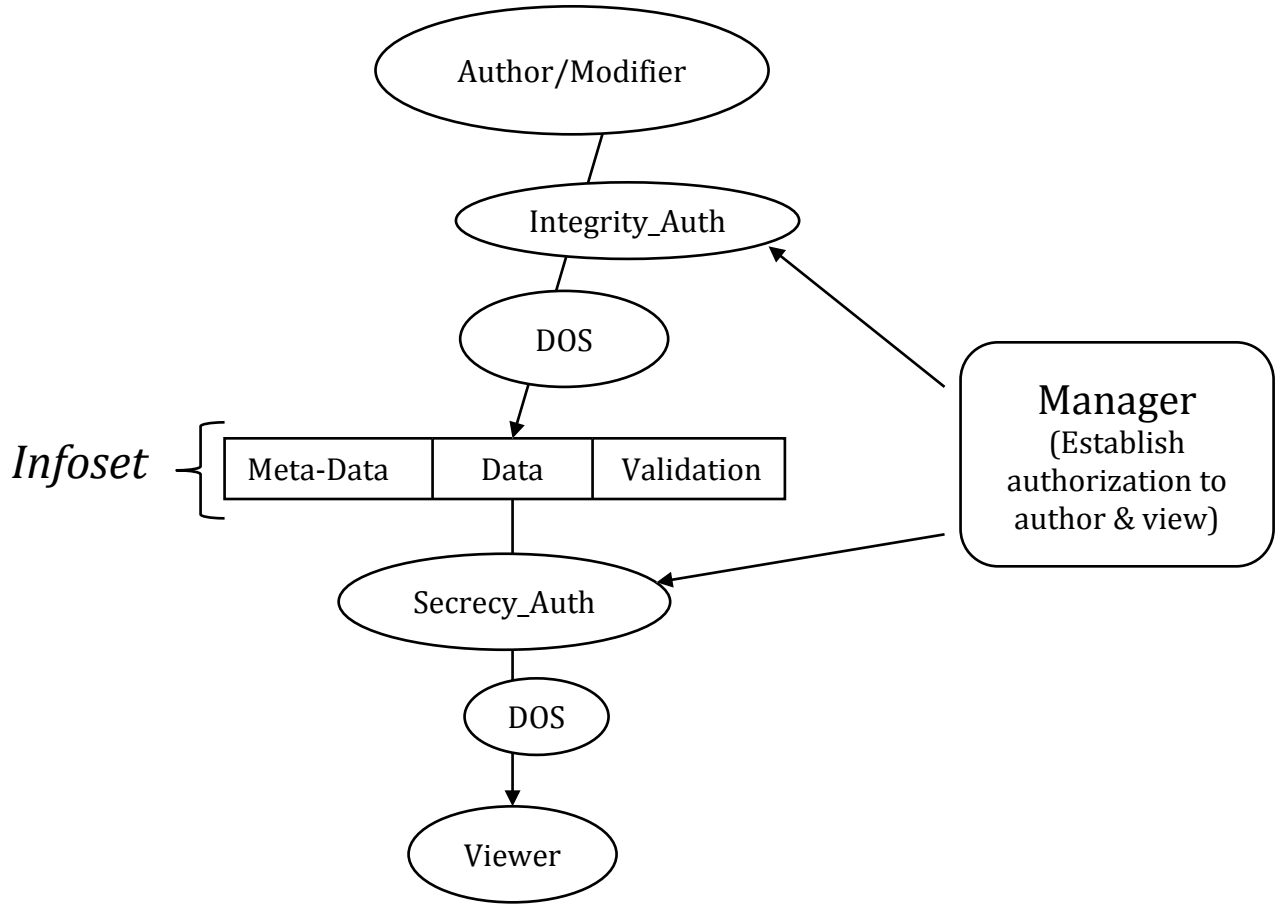

Figure 1. An Infoset and relationships to Author and Viewer.

Secrets

A secret is defined by Merriam-Webster[14] as something kept hidden or shared only in confidentially with a few. In the realm we are concerned with in this paper, the digital world, a secret can be defined to be any information (information being a well described or delineated set of "bits") kept hidden from others or known only to oneself or to a few. Usually, there is some reason behind keeping the information from a broader audience.

If the existence of a secret is known (from the metadata), then multiple parties may guess or speculate regarding the secret information, but protections are put in place to assure that the secret is known with high certainty only by oneself or by a few. The robustness with which a designated secret is protected is called its secrecy[15] (high, medium, low), and should reflect the probability that an adversary can gain knowledge of the secret.

[14] http://www.merriam-webster.com/dictionary/secret

[15] Another common notion of "secrecy" is not robustness of protection but simply the extent to which a secret is known. A secret known by only one person has more secrecy than a secret known by five people, for example.

The threats to the integrity of the infoset center around robust authentication and authorization of entities who are intended to author (modify) the data, and also around the adversary's ability to spoof a viewer into viewing a bogus infoset, and of the adversary's ability to deny access to the infoset on the part of the author or of the viewer.

The threats to the secrecy of the infoset center around robust authentication and authorization of entities who are intended to view the data, and also around the authentication and authorization of entities who are intended to author/modify the data, as well as the adversary's ability to deny access to the infoset on the part of the author or of the viewer.

One of the confusing things about integrity and secrecy is that integrity can be considered separately from secrecy, but secrecy cannot be considered without integrity. That is, the integrity of a publicly accessible infoset is independent of viewer, while the integrity of a secret infoset depends on the authorization of the viewer (the infoset must have high uncertainty therefore low integrity to unauthorized viewers, while maintaining low uncertainty and high integrity to authorized viewers). In spite of this dependence, it is useful to separate the concepts of integrity from secrecy in security systems, since (for reasons cited throughout this report) it is easier to protect integrity of an infoset than it is to protect its secrecy.

Integrets

For the purposes of this paper, we define an "integret" (noun) to be information (again a well described or delineated set of "bits") that is shared between users and known with high certainty by many parties. The existence of the integret is known and protections are put in place to assure that the integret is known with high certainty by many or all parties. The robustness with which a designated integret is protected is called its integrity (high, medium, low), and should reflect the probability that an adversary can reduce the certainty with which the multiple parties know the (true) integret. Protecting a secret requires physical barriers. Protecting an Integret typically requires comparison with multiple copies held in a manner that the majority of copies are difficult for an adversary to subvert simultaneously.

Protection for an Integret

An *Integret* is an information set created by an author or by a well defined relationship to another information set that is shared between users and known with high certainty by many or all parties. To protect the integrity of an integret, these methods can be used:

1) place replicas in multiple places (so that it is hard to modify all copies) & "vote" by using the value of the majority of copies that still match to ascertain the unmodified infoset.

2) place physical and/or logical barriers between the modification of the infoset and unauthorized modifiers

3) embed a checksum with the infoset that can be recalculated to detect modification by use of a secret key

a. symmetric key (calculation is fast, requires protection of multiple copies of the secret key, and there is no protection against non-repudiation among holders of the multiple secret key copies,)

b. asymmetric key (calculation is slower, but protects against non-repudiation, since it requires only a single secret (per author) to protect.

Protection for a Secret

Now consider typical protections for a secret:

1) place physical and/or logical barriers between the infoset and unauthorized viewers

2) encrypt the infoset's data and possibly its metadata so that the infoset can be decrypted (revealed) only by those who possess the right secret key. This encryption can be accomplished with these techniques

    a. Symmetric key (efficient and fast but requires secure exchange of secret key known to both ends of the communication. If the key is as long as the message, this results in perfect secrecy <in the "Shannon" sense>, and is called a "one-time-pad".)

    b. Asymmetric key (slow but does not require apriori exchange of secret information to enable confidential communication)

    c. hybrid (Asymmetric key encryption for key management or key exchange, then faster Symmetric key encryption for protection of the bulk of the secret data.

    d. Quantum Cryptographic Techniques (Quantum Key Distribution[16] is used to exchange enough key material to complete the confidential transfer using symmetric encryption.)

In spite of the use of these secret protection techniques,:

- Secret keys can be compromised by an "insider".
- Secret keys can be compromised by inadvertent disclosure or poor operating practice.
- Secret keys can be compromised by poor implementation of key generation and distribution protocols
- Secret keys can be extracted by sophisticated reverse engineering and circumvention of barriers against such extraction

[16] BB84 is a quantum key distribution scheme developed by Charles Bennett and Gilles Brassard in 1984. It is the first quantum cryptography protocol. The protocol is provably secure, relying on the quantum property that information gain is only possible at the expense of disturbing the signal. It is used as a method of securely communicating a private key from one party to another for use in one-time pad encryption. See C. H. Bennet and G. Brassard, "Quantum Cryptography: Public key distribution and coin tossing", in Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing, Bangalore, p. 175 (1984)

- Secret key bits are continually “leaked” through “side channels (susceptible to power analysis, for example[17])

The “First Principles” regarding resiliency of Authentication Systems against catastrophic failure due to compromise of secrets used for authentication deduced from this analysis are:

- The fewer secret keys involved in an authentication operation, the simpler and more reliable the authentication can be. (especially in high exposure environments)
- If secret keys cannot be eliminated from an authentication design, then these keys should be designed to reside in portions of the life cycle that are easier to protect.
- Secret keys may be required to protect confidentiality of data, but there are more design choices to protect these keys if authentication of data is accomplished without dependence on small secrets.

[17] In cryptography, a **side channel attack** is any attack based on information gained from the physical implementation of a cryptosystem, rather than brute force or theoretical weaknesses in the algorithms. For example, timing information, power consumption, electromagnetic leaks or even sound can provide an extra source of information which can be exploited to break the system. Many side-channel attacks require considerable technical knowledge of the internal operation of the system on which the cryptography is implemented.

"Authentication without Secrecy" is not the same as "Authentication without Secrets"

In the literature "authentication without secrecy" (originally studied by Sandia Fellow Gus Simmons[18]) means a transmitter sending a message to a receiver over a publicly exposed channel such that the content of the message is not hidden[19]. This body of work was motivated by the goal of nuclear treaty verification, in which sensors emplaced in multiple countries to detect nuclear tests must not communicate secret information but must be verifiable by multiple parties that the data is authentic. It deals with preventing the adversary from injecting a fraudulent message (impersonation) or modifying an intercepted one (substitution), while making the receiver believe that the message is authentic. An authentication code A is a class of invertible functions A_h^{XY} that generates a codeword $y \in Y$ for a message $x \in X$ and is indexed by the key information h. Perfect protection is defined to be achieved if the best strategy of the adversary in an impersonation or substitution attack is random selection with uniform distribution from the set of possible messages.

No cryptographic methods exist to authenticate messages without relying on protection of a small secret key somewhere in the operation. Current authentication practice falls into two cryptographic categories (and a hybrid combination of them), and a "biometric" category that does not rely on protection of secrets but on the difficulty of reproducing certain complex characteristics of the originator.

Shared Secret Key Authentication- Many modern crypto methods implement "symmetric secret key" authentication, in which a hash of a message is encrypted by a shared secret key, and if decrypted and checked, then the message body (sent in the clear, therefore qualifying as "authentication without secrecy") is deemed authentic. Shared symmetric secret key authentication cannot protect against non-repudiation, as any party sharing the secret key can fabricate a message regarded as authentic. (For example, in a scheme in which only two parties share the secret and can encrypt and validate each other's messages, the receiver can fabricate a message that it fraudulently asserts came from the sender.)

Asymmetric Key Authentication- An advance in common use in which messages are signed via a secret key but are validated via a public key, thereby eliminating the need to protect a secret during the validation process and providing a method to achieve non-repudiation between sender and receiver.

Biometric Authentication - A third method, generally not considered to be "cryptographic" consists of recognizing unique characteristics of the originating person or his/her imprint on the message. These techniques include fingerprint or recognition of other personal characteristics, analysis of handwriting characteristics, and voice recognition for spoken messages. It is this class of authentication that is the most intriguing to this study, as it serves as an example of authentication without secrets (since the characteristics of an individual or his/her imprint on a

[18] G. J. Simmons, "Message authentication without secrecy," in AAAS Selected Symposia Series (G. Simmon, e d .) , pp. 105-139, 1982.
[19] http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=695179&isnumber=4025

message are not secret, but merely hard to reproduce except in the form of an exact photograph or recording that is hard to apply to a different message.

Message Authentication

In this section, we are concerned with how well we can trust the information we have received. Actions based on bad (or false) information can have disastrous consequences. Typically, we use secrecy to protect disclosure of information from those who do not have a need to know. It would be easy to erroneously rely on secrecy to achieve some measure of authentication. For instance, if you have an encrypted document (a file), and were given the key by the supposed author to open the document, it is easy to make the assumption that the information is authentic and has not been modified because you can successfully decrypt the file (if not authentic, the result of some kinds of decryption may result in random garbage easily discernable from the expected information) [20]. There are other reasons not to rely on symmetric encryption and a *secret* key to provide authentication. In particular, this example does not achieve non-repudiation, as the authenticating party now holds the same secret key used to encrypt the file and can re-encrypt a modification that will also decrypt properly with the same key.

Imagine a letter, written by one person to another. In the typical parlance of "crypto talk" let's call them Alice and Bob. Alice's letter may be conveyed from sender to receiver through an unsecured channel (via a courier, we will call him Charlie). The lack of secrecy would be the equivalent of transporting the letter without an envelope. Charlie, or anyone else for that matter, can read the contents of the letter, make a copy, transmit it to other parties, etc. However, once delivered, Bob might or might not be able to determine that the letter was authentic (from the actual sender, Alice). How does Bob know that the letter wasn't opened, modified and resealed by Charlie? Under these circumstances, he doesn't. What are the ways that the receiver would authenticate the authorship of the letter and thus authenticate the message. Perhaps only Bob would recognize the handwriting, the style of the letter or the paper it was written on. This would all be under the assumption that the contents of the letter was in plain text and not encrypted with some previously defined code. Then again, the particular use of language is in effect a code identifying the authorship. This would require prior communications with the sender Alice to establish familiarity between Bob and Alice.

Someone familiar with the particular way the author writes (or speaks) would be able to say with some certainty (not absolute certainty) that the letter was written by Alice. But again, the knowledge of a writer's style is not really a secret. There is no need to keep this information secret, just to have the information on hand, or in other words have some knowledge of the author. The more knowledge Bob has of the author's background, traits, writing style and other personal characteristics, the better an authentication can be performed (and the more easily Bob can forge a message that will falsely authenticate as originating from Alice, unless the personal characteristics are hard for Bob to reproduce). Study of these characteristics is exactly what has been done by Bible scholars trying to authenticate the Dead Sea Scrolls.[21] As is often the case,

[20] We note that not all encryption systems are engineered to have this property and "encryption" algorithms in general are a poor substitute for cryptographic algorithms designed for "authentication".
[21] Palaeography (the study of ancient writing, shape and style of the letters, etc.) has been used to date the Dead Sea

the characteristics that are used to provide authentication have nothing to do with the actual message itself.

But what about a letter sent to someone without intimate knowledge of author? How does the recipient authenticate the letter? Historically, this problem was solved by sealing the letter and putting an emblem on the seal that only the sender would have in their possession. This provided some measure of protection of the contents from modification and at the same time provided authentication that the letter was written by the author and not a forgery. Later, signatures became a way to provide authentication to the letter. One's signature was considered unique. The signature is not a secret, it can be provided to everyone for them to use in comparison to the specific letter they are authenticating. This does not provide protection from forgery. It does however require someone, a trusted third party "signature expert", to verify that the signature is authentic. In the next section we will examine this process of providing authentication of one's identity.

Authentication and Identity

Authentication (from the Greek word *αυθεντικός*, meaning real or genuine) is the process or act of determining whether someone or something is authentic, or who or what it is declared to be[22]. In the current case, it refers most often to the process by which one uses an authentication factor to provide proof of an identity to a data processing system.

Authentication is the process of identifying an individual, usually based on a username and password[23]. In security systems, authentication is distinct from *authorization ,* which is the process of giving individuals permission to access system objects based on their identity. Authentication merely ensures that the individual is who he or she claims to be, but says nothing about the rights of the individual to access a specific piece of information.

Scrolls. (http://www.apologeticspress.org/articles/266)

[22] SearchSecurity.Com Definitions: http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci211621,00.html and http://en.wikipedia.org/wiki/Authentication

[23] http://www.webopedia.com/TERM/A/authentication.html

As mentioned previously, authentication usually involves three authentication factors[24]:

Ownership Factor
Commonly referred to as *something you have*, these factors could include a multitude of common items including an ID card, token or smart card. .

Knowledge Factor
This commonly refers to *something you know* and might be something like a password, personal PIN number or pass phrase. These codes are often a set of alpha-numeric digits usually created following specific rules but easily remembered by a human.

Inherence Factor
This commonly would be *something you are or do*, that is something that is uniquely (and inherently) yours and is tied to your person. The most familiar of these, such as the fingerprint or retinal pattern, are in use today. However, recent advances in science and data processing has made it possible to use facial recognition or voice recognition and in the future perhaps DNA sequence could be instantly analyzed. It is possible that many other unique biometric signatures will be discovered.

In private and public computer networks (including the Internet), authentication is commonly done through the use of logon passwords. Knowledge of the password is assumed to guarantee that the user is authentic. Each user registers initially (or is registered by someone else), using an assigned or self-declared password. On each subsequent use, the user must know and use the previously declared password. The weakness in this system for transactions that are significant (such as the exchange of money) is that passwords can often be stolen, accidentally revealed, forced from the person using duress, or forgotten.

For this reason, those participating in Internet business and many other transactions require a more stringent authentication process. The use of digital certificates containing public keys issued and verified by a Certificate Authority (CA) and incorporated into a public key infrastructure is considered likely to become the standard way to perform authentication on the Internet. The authentication itself involves interactively providing a "response" that requires decryption of a "challenge" by the entity being authenticated using a uniquely held private key.

[24] http://en.wikipedia.org/wiki/Authentication

Schneier in "Applied Cryptography" goes into great detail about proofs of identity.[25] Here is a short quote from his book.

> "In the real world, we often use physical tokens as proofs of identity: passports, driver's licenses, credit cards, and so on. The token contains something that links it to a person: a picture, usually, or a signature, but it could almost as easily be a thumbprint, a retinal scan, or a dental x-ray. Wouldn't it be nice to do the same thing digitally?"

Schneier then describes how Uriel Feige, Amos Fiat, and Adi Shamir proposed using zero-knowledge proofs as proofs of identity.[26,27] However, for the purposes of this work, their solution involved the creation of a private key. Using a zero-knowledge proof, Alice proves that she has knowledge of her own private key by signing using it and Bob can use Alice's public key to verify her identity. This system has its flaws, which have been described by researchers. However, each scheme provides some measure of security in the difficulty in performing some particular algorithm without knowledge of the authentication secret.

Two Factor Authentication

Two-factor authentication[28] is a security process in which the user provides two means of identification, one of which is typically a physical token, such as a card, and the other of which is typically something memorized, such as a security code. In this context, the two factors involved are sometimes spoken of as *something you have* and *something you know*. A common example of two-factor authentication is a bank card: the card itself is the physical item and the personal identification number (PIN) is the data that goes with it.

According to proponents, two-factor authentication could drastically reduce the incidence of online identity theft, phishing expeditions, and other online fraud, because the victim's password would no longer be enough to give a thief access to their information. Opponents argue (among other things) that, should a thief have access to your computer, he can boot up in safe mode, bypass the physical authentication processes, scan your system for all passwords and enter the data manually, thus -- at least in this situation -- making two-factor authentication no more secure than the use of a password alone.

---

[25] Bruce Schneier, "Applied Cryptography, Second Edition," John Wiley & Sons, Inc., New York, 1996.
[26] A. Fiat and A. Shamir, "How to Prove Yourself: Practical Solutions to Identification and Signature Problems," *Advances in Cryptography* – CRYPTO '86 *Proceedings*, Springer-Verlag, 1987, pp. 186-194.
A. Fiat and A. Shamir, "Unforgetable Proofs of Identity," *Proceedings of Securicom* 87, Paris, 1987, pp. 147-153.
U. Feige, A. Fiat, and A. Shamir, "Zero-Knowledge Proofs of Identity," *Proceedings of the 19th Annual ACM Symposium on the Theory of Computing*, 1987, pp. 210-217.
U. Feige, A. Fiat, and A. Shamir, "Zero-Knowledge Proofs of Identity," *Journal of Cryptography*, v. 1, n. 2, 1988, pp. 77-94.
[27] The details of this patent and how it was determined to be "detrimental to the national security" can be found in S. Landau, "Zero-Knowledge and the Department of Defense," Notices of the American Mathematical Society, v. 35, n. 1, Jan 1988, pp. 5-12.
[28] http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci992919,00.html

Some security procedures now require *three-factor authentication*, which involves possession of a physical token and a password, used in conjunction with biometric data, such as finger scanning or a voiceprint.

In 2005, NIST introduced FIPS PUB 201 which describes a smart card system for Personnel Identity Verification that is capable of enabling three-factor authentication. While these "PIV-201" cards (currently replacing cleared personnel badges throughout the government) are able to store several private keys for different purposes (cryptographic authentication, cryptographic encryption, maintenance of the card, etc.), these cards can contain biometric reference data for an individual and a pin or password to enable its operation. Even though we currently do not utilize all the capabilities of these cards, there is the potential to use "something you know" to unlock/enable the card, "something you have" in the form of the card authenticated via its private key, and "something you are" in the form of comparison of a real-time biometric measurement with the biometric reference data signed and stored on the card. In general, this represents a great advance in authentication technology (to simultaneously utilize all three factors), but these cards still rely on a secret to enable the card, and an embedded private key to authenticate the card, and a secret somewhere in the system with which to sign the biometric reference data.

Authentication and Authorization

Logically, authentication precedes authorization (although they may often seem to be combined).

Authentication is the process of confirming the identity of a person that is attempting to access a system or of confirming the *authenticity* of a message[29].

Authentication is distinct from *authorization*, which is the process of giving individuals access to system objects based on their identity. Authentication merely confirms the identity of the individual, but says nothing about its access rights. Authenticity refers to whether both the source and the content of a message are what they are claimed to be.

Authentication can be based on something that a person knows, has or is (inherency). Examples of the first include user names, passwords and pass phrases. Examples of the second include IP addresses, digital signatures, cell phones and identification cards. The third consists of biometric data, which includes fingerprints, palm patterns, iris scans, voice recognition and facial recognition.

A *digital signature* is a mathematical method for authenticating digital information which is implemented using techniques from public key cryptography (PKC). It usually involves two complementary algorithms, one used for signing and the other used for verification. The digital signature provides the recipient some level of assurance that the message that was received (or document) was actually sent by the sender and that the message has not been altered in transient.[30] Menezes refers to this as data origin authentication and data integrity.[31]

---

[29] http://www.bellevuelinux.org/authentication.html
[30] http://en.wikipedia.org/wiki/Digital_signature#Some_digital_signature_algorithms

There are a number of algorithms that have been put forward in support of digital signatures. Currently, the NSA has approved Elliptic Curve Digital Signature Algorithm (ECDSA) – FIPS PUB 186-3 (using the curves with 256 and 384-bit prime moduli) for digital signature standard (DSS) as part of Suite B set of algorithms for protection of national security information.[32] Like all other algorithms before it, this one requires the generation of a private key/public key pair and of course the protection of the private key.

None of these methods are completely secure, and all could be vulnerable to *spoofing*, i.e., pretending to be someone or something else. For example, there are ways of discovering user names and passwords, IP addresses can be forged, and even fingerprints can be falsified (such as by using a thin layer of a transparent material that contains someone else's fingerprints).

The chances of successful unauthorized access can be greatly reduced by requiring multiple types of authentication.

Quantum Authentication

As Quantum Key Distribution protocols are beginning to be used to protect the confidentiality of data[33], some researchers have begun to look into what it means to provide authentication of quantum messages[34]. In the classical sense, protecting a message from modification and providing positive authentication is a cryptographic function. The strength of the authentication protocol depends mostly on selection of the encryption function.

Standard classical authentication techniques no longer work when sending quantum information. In the traditional communications example (Alice sends a quantum message to Bob, and this message is being intercepted by Eve) applied to quantum messages, when Eve reads the message, the quantum state is altered. Therefore, the authentication protocol must protect superposition of states (quantum state $|0\rangle + |1\rangle$ is different from the quantum state $|0\rangle - |1\rangle$).

Gottesman describes the protection of quantum states from the man-in-the-middle attack. A quantum authentication protocol must encode the quantum state in a "quantum error-detection, error-correcting code." An error detection code would work because Bob only cares about detecting modifications to the authentication by Eve. Alice and Bob must also encrypt the quantum state. In the classical authentication case, Eve can read the message, as long as she does not change it (thus betting caught). In the quantum case, just the act of reading or measuring the state by necessity changes it.

Barnum, et al, describe in detail a scheme for the authentication of quantum messages[35]. They set forth a formal definition of authentication for quantum states, construct efficient purity testing protocols and propose the impossibility of digitally signing quantum states. This authentication scheme relies on classical keys instead of entangled quantum keys.

---

[31] Alfred J. Menezes, et al, "Handbook of Applied Cryptography," CRC Press, 1997, p. 359.
[32] http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf
[33] http://www.eetimes.com/story/OEG20021111S0036
[34] White paper, arXiv:quant-ph/0205128v1
[35] H. Barnum, et al, "Authentication of Quantum Messages," 20 May 2002, (arXiv:quant-ph/0205128v1).

One of the most interesting findings in this work is the inability to separate authentication from confidentiality in quantum information systems. In classical systems, one can send a message in the clear, with a cryptographic checksum bound to it that can prove the authenticity of the message. In a quantum system, since the evesdropper is unable to copy the message during transmission, quantum authentication schemes also provide (and are inseparable from) quantum encryption. The implications of this dichotomy between classical information processing and quantum information processing deserves more study.

Digital Signatures

Recall that Alice signed her letter and that her signature is uniquely her's and not easily forged. The *digital signature* is the digital counterpart to Alice's hand written signature. The concept of a digital signature was first introduced by Diffie and Hellman in 1976. By 1981, when Diffie wrote his 15 year forecast, research into public key encryption was already reaching a broad audience.[36] The digital signature is a calculated number dependent on some secret known only to the signer, Alice, and dependent on the content of the message being signed.[37] Digital signatures can provide authentication, data integrity and non-repudiation and certification of public keys. There are multiple signature schemes all having the same characteristic; they require the signer to know (and keep) a small secret. It is knowledge of this secret that provides the signer the ability to prove they created the message. However, there must be a way for Bob to prove that Alice was the only one that could have possibly signed the message. This is where public-key private-key cryptography came to the rescue and has formed the basis for a multitude of digital signing schemes.

Digital Signature Algorithm

The U.S. National Institute of Standards and Technology (NIST) proposed a digital signature algorithm (DSA) in 1991. Since then, DSA has become a U.S. Federal Information Processing

[36] In his paper "Cryptographic Technology: Fifteen Year forecast," Whitfield Diffie looks into his crystal ball at the "future" of cryptography based on current events of 1981. This paper presents a very good overview of the early efforts in public key encryption and is still a valuable resource in 2009. Originally prepared under contract to CRC Systems, this paper was reprinted in a AAAS Selectted Symposia in 1982. Whitfield Diffie, "Cryptographic Technology: Fifteen Year Forecast," AAAS Secure Communications and Asymmetric Crypto Systems, Ed. G.J. Simmons, vol. 69, Westview Press, boulder, Colorado, 1982.

[37] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, "Handbook of Applied Cryptography," CRC Press, 1996.

Standard (FIPS 186) May 19, 1994 and is formally called Digital Signature Standard (DSS). From the standard:[38]

> **Explanation:** This Standard specifies a Digital Signature Algorithm (DSA) appropriate for applications requiring a digital rather than written signature. The DSA digital signature is a pair of large numbers represented in a computer as strings of binary digits. The digital signature is computed using a set of rules (i.e., the DSA) and a set of parameters such that the identity of the signatory and integrity of the data can be verified. The DSA provides the capability to generate and verify signatures. Signature generation makes use of a private key to generate a digital signature. Signature verification makes use of a public key which corresponds to, but is not the same as, the private key. Each user possesses a private and public key pair. Public keys are assumed to be known to the public in general. Private keys are never shared. Anyone can verify the signature of a user by employing that user's public key. Signature generation can be performed only by the possessor of the user's private key.
>
> A hash function is used in the signature generation process to obtain a condensed version of data, called a message digest (see Figure 1). The message digest is then input to the DSA to generate the digital signature. The digital signature is sent to the intended verifier along with the signed data (often called the message). The verifier of the message and signature verifies the signature by using the sender's public key. The same hash function must also be used in the verification process. The hash function is specified in a separate standard, the Secure Hash Standard (SHS), FIPS 180. Similar procedures may be used to generate and verify signatures for stored as well as transmitted data.

The purpose of a digital signature is to provide proof that the message has not been altered in transit and to provide certainty of the originator's identity. DSA makes use of public/private key encryption. The private key is used in the signature generation process and the public key is used in the signature verification process. Without the private key, an adversary cannot generate the correct signature, i.e. signatures cannot be forged. Anyone can use the signatory's public key to verify the signature.

The DSA algorithm relies upon proper generation of two large prime numbers. They are referred to as p and q and are defined in FIPS 186 as:

p = a prime modulus, where $2^{L-1} < p < 2^{L}$ for $512 \le L \le 1024$ and L a multiple of 64

q = a prime divisor of $p - 1$, where $2^{159} < q < 2^{160}$

[38] http://www.itl.nist.gov/fipspubs/fip186.htm

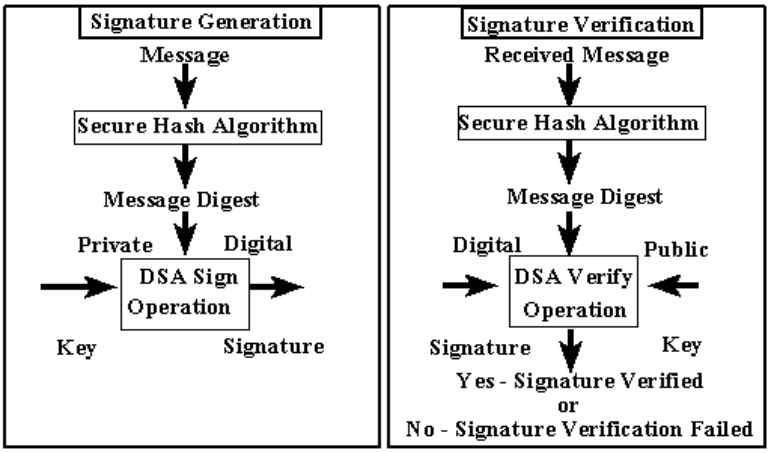

Figure 2. Using the SHA with the DSA (from NIST).

The process for generation of these prime numbers is spelled out in great detail in FIPS 186. We cover the highlights here for the purposes of illustration. This process requires choosing an arbitrary sequence of 160 bits called the SEED and computing

$$U = SHA[SEED] \text{ XOR } SHA[(SEED+1) \bmod 2^g]$$

Where g is the length of SEED in bits. The key process is generating the SEED and keeping this value secret. Next we form q from U by setting the most significant bit and the least significant bit to 1. Test whether q is prime (as required by the standard). Once you have a prime number q, we start the process of generating p by an iteration process and checking the value of p until it matches the criteria.

Communications Between Components

Every system is made up of smaller individual components. A radio might have three modules each having multiple printed circuit boards and on the PC boards are integrated circuits performing tasks. Even the radio is a component of a larger system, say an airplane and it communicates with other devices on the plane. At each level of this system, we find communications. The purpose of these communications are many including command and control, passing status information and of course passing data. In order to secure the communications, we encrypt. To encrypt we insert a secret inside the component and this secret becomes the basis for the encryption key. In the case of public/private cryptography, this secret is used to generate unique private key and public key.

In order to protect the secret, we provide barriers. The barriers add to the complexity of the component as well as the cost. And should the barriers prove inadequate, the secret is lost and communications (or authentication) is compromised.

What if it were possible for two components to communicate and authenticate each other without the use of a secret (key)? There are many advantages to this type of technique including reduced manufacturing cost and no risk to the exposure of the secret key (because there is none!).

Authenticating Components

If we think in terms of authenticating components of a larger system rather than messages, the situation changes very little. Let's compare the two. A component has a creator/modifiers (messages have authors and editors), and users (readers). How can a user ascertain that a component was created by the expected creator and is unmodified except by authorized modifiers? Let's look at some possible ways to accomplish this task.

1) **Compare -** grant the user detailed knowledge of the as-built design and the user compares the component to design, looking for discrepancies
2) **Barrier -** place barrier between component and unauthorized modifiers
    a. make barrier transparent but hard to breach
    b. make barrier secret and hard to breach (adversary first has to discover nature of barrier)
3) **Secret -** embed secret in component that can be used in zero-knowledge protocol to prove existence of secret within component (place barrier between secret and unauthorized modifiers or readers, but this barrier is smaller and easier to manage than the larger barrier in (2) above.
4) **Measure –** Measure some characteristic of the component that is difficult to reproduce or clone in another component (PUF as fingerprint, etc.)

Perhaps component-to-component or component-to-system familiarity can be used to provide authentication. Simply being able to recognize a component as the same component seen before (in manufacturing, for example) is insufficient to assure against modification in the manufacturing portion of the life cycle, but would assure against substitution and/or forgery later in the life cycle.

Let's assume that a component is placed into a system in a secure location. The component then establishes communication channels between itself and other components to which it is connected. In this situation, the components can establish a relationship that can be used to provide for authentication throughout the lifetime of the component. That relationship can take the form of secret keys that are passed and used for symmetric encryption or (even better) public keys that are used to establish authentication. In either case, there is a secret that must be protected from disclosure. However, if there were a metric of uniqueness inherent to the component then this could be measured and used much like a biometric is used to authenticate human individuals.

Here we enumerate these possibilities.

1) by special mark (analogous to symmetric secret key methods)
2) how would one form a “checksum” over the “as-built” design that would include artifacts not described in the design (extra antennas, etc.)
3) “biometric” measurement of a quality unique to the device
    a. This could be done in a destructive fashion, for example, by examining photomicrographs of small variations in the manufacturing process of integrated circuits after de-lidding the package and destructively removing some layers of the fabricated part.
    b. Means to measure unique characteristics can be incorporated into the design of the device. These means can then be used to report the measurements in a non-destructive fashion, but must also be protected against subversion and the external infrastructure that processes the measurement must be protected against being spoofed into thinking that a playback of a previous measurement is a new real-time measurement. The basis for measuring unique characteristics of a device called “Physically Unclonable Functions” (PUF) is explored in the next section.

BIOMETRIC MEASUREMENTS AND PHYSICALLY UNCLONABLE FUNCTIONS

Biometrics

This section examines biometrics as an example of authentication without secrets and provides an analysis of the difficulty of differentiating between a real-time "biometric" measurement and a playback, etc.

In order to accomplish authentication without secrets, we have examined the biometric model for authenticating humans. The elements of the biometric authentication system are shown in Figure 3. Biometric systems depend on a high-integrity measurement of one or more hard-to-clone human characteristics, and an ability to differentiate between a real-time measurement and a playback of such a measurement (the ability to differentiate between a photograph of a face and a real face in facial recognition, for example). The differentiation between the real-time measurement of the unique characteristic and a playback is made difficult if attempted remotely.

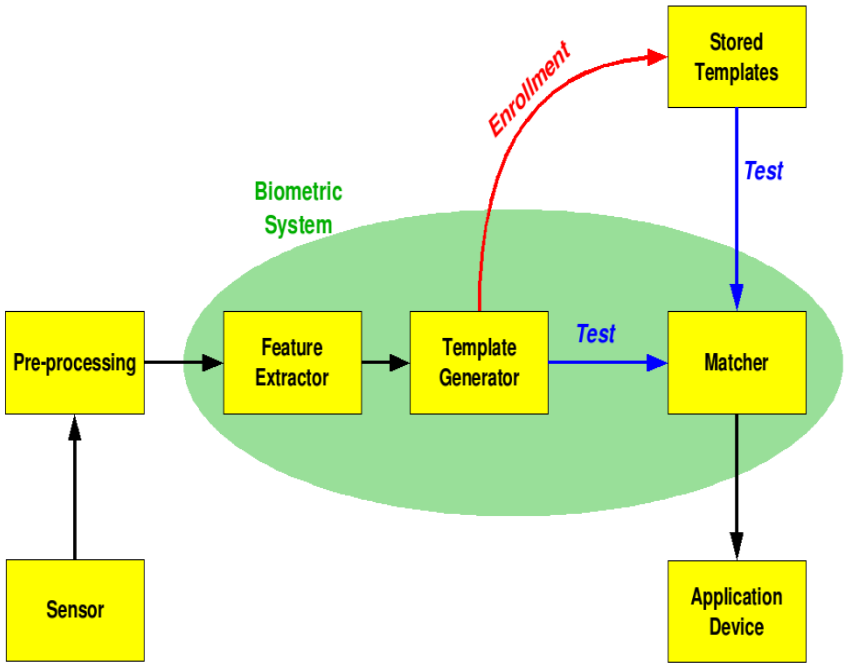

Figure 3. Elements of a Typical Biometric Authentication System.

A fingerprint reader is relied upon only in combination with other authentication methods, or if by local observation (human supervision or attendance of a fingerprint reader for example) to assure against a "man-in-the-middle" attack that would somehow insert an image of the desired fingerprint (a playback) rather than the real-time measurement. Providing assurance is made even more difficult by remote operation. This is demonstrated in Figure 4 where the man-in-the-middle can execute a playback attack.

One approach to performing a remote fingerprint authentication measurement would be to somehow enclose within a protected volume a high-integrity reader, the reference fingerprint data and the comparison mechanism, along with a message signing mechanism intended to protect the authentication comparison message over the communication channel. But the only known message authentication methods for the communication channel involve the introduction of secret cryptovariables and the brittleness we are trying to avoid.

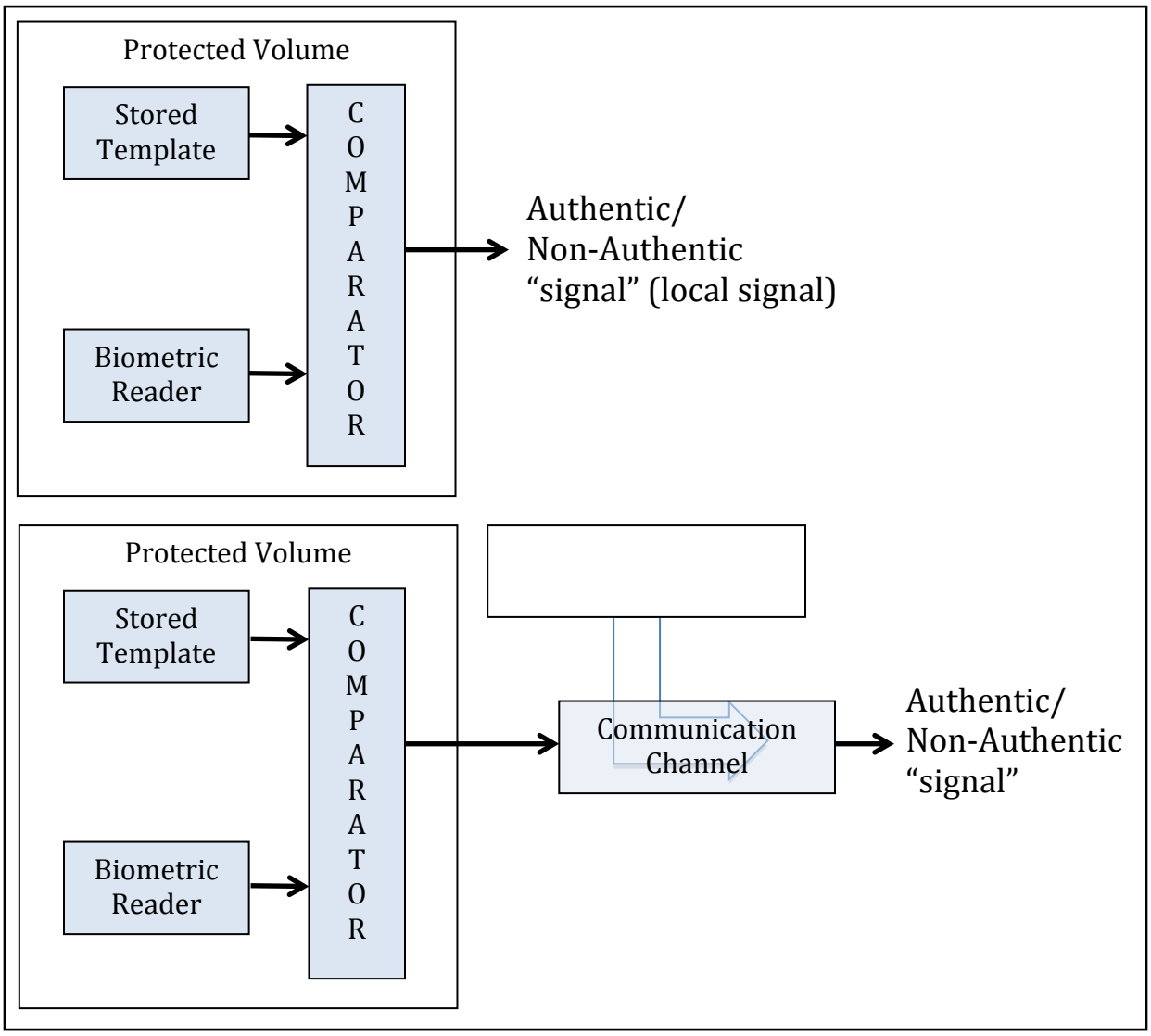

Figure 4. Remote Biometric Authentication.

Physically Unclonable Functions (PUFs)

In practical cryptography, a **PUF** or **Physically Unclonable Function**[39] is a function that is embodied in a physical structure that is easy to evaluate but hard to characterize.

The physical structure that contains the PUF consists of many random components. These random components are introduced during the manufacturing process and cannot be engineered to behave in an exactly predictable manner.

Some PUFs exhibit natural challenge-response behavior. When a physical stimulus is applied to the structure, it reacts in an unpredictable way due to the presence of the randomness in these components. The applied stimulus is called the challenge, and the reaction of the PUF is called the response. A specific challenge and its corresponding response together form a challenge-response-pair or CRP. Other PUF mechanisms have no clear input or "challenge". In such cases, PUF measurements are used without a challenge-response protocol, or the PUF measurement can be used as a seed for generation of an "extrinsic" private key-public key pair which can be used in a cryptographic challenge-response protocol. A separate LDRD has thoroughly examined PUFs and infrastructure required to "fingerprint" integrated circuits. See "Infrastructure for Nondestructive, Real-Time Fingerprinting of Integrated Circuits, by Jason Hamlet, Todd Bauer, and Lyndon Pierson, SAND2009-TBD, therefore similar material on PUFs will not be reproduced here.

PUFs inherit their unclonability property from the fact that every PUF has a unique and unpredictable way of mapping challenges to responses. Two PUFs that were manufactured with the same process will still possess a unique challenge-response behavior. It is very hard to construct a PUF with the same challenge-response behavior as a given PUF. Physically reproducing the same response in multiple devices is very hard because exact control over the manufacturing process, such that all parameters of the physical structure can be exactly defined, is very hard. Mathematical unclonability means that it should be very hard to compute an unknown response given the exact parameters or other CRPs from the PUF. This is because a response is created as a very complex interaction of the challenge with the random components. Modeling this interaction, even if the random values are known, should take a lot of computational effort. The combination of physical and mathematical unclonability renders a PUF truly infeasible to reproduce, and if designed properly, the response can be differentiated from a playback of a previous response.

[39] http://en.wikipedia.org/wiki/Physically_Unclonable_Function#cite_note-0

Even so, the PUF measurement resembles a secret which, if it becomes known to the adversary, could be used to spoof the authentication. In practice, PUF measurements are regarded as secrets that are difficult to extract without access to the on-chip measurement circuitry itself. Two modes of authentication using PUFs present themselves; 1) the inherent (natural) PUF CRPs must be initially measured by a trusted third party in a controlled environment (so that subsequent CRPs can be sent to this trusted party for verification), or 2) PUF measurements can be processed into "extrinsic" private keys that can be used in a cryptographic challenge-response protocol that eliminates the need for direct interaction with a trusted third party for each validation. In either of these cases, there is need for error correction to cope with noise inherent in multiple PUF measurements within the same device[40].

It is easy to envision the design of mechanisms to leverage a PUF into measurable characteristics that are hard-to-clone, but not hard to "play back". It is the authentication system requirement for differentiation of a real-time characteristic measurement from a playback of a previous measurement that necessitates an interactive challenge-response or equivalent protocol.

[40] In some papers this noise is called "intra-device variation" to distinguish it from the variation intended from device to device called "inter-device variation".

METRICS FOR BRITTLENESS OF AUTHENTICATION SYSTEMS USING SECRET KEYS

Brittleness or fragility of authentication is due to incorporation of secrets that if exposed, render the authentication useless because the adversary would then have the means to spoof the authentication. The opposite of this brittleness is *resilience* to catastrophic loss of authentication capability due to failure of secret-keeping mechanisms.

Development of a metric for brittleness of authentication systems will enable a more systematic study of authentication protection mechanisms and an ability to compare different authentication systems to each other.

The development of such a metric is complicated by the difficulty of assessing the effectiveness of "anti-reverse-engineering barriers" used to prevent the extraction of secrets from the hardware or software required to store and process them in the course of the authentication. Experts interviewed in the course of this work agree that these "anti-reverse-engineering barriers" (e.g. protective coatings designed to destroy secrets if the coating barrier is breached, etc.) are merely designed to increase the adversary's cost and to delay the adversary's extraction, but these so-called barriers are effective only in low exposure environments in which the adversary has little or no access to a sufficient number of units for reverse engineering. These barriers add little or no value in a high exposure environment where a motivated adversary can bring large resources to bear to breach the protection mechanism.

Another difficulty for brittleness metrics is the variability of exposure to the adversary in various parts of the life cycle. Secrets may be protected in a guarded facility in one phase of the life cycle, and protected in a standard, commercially available smart card in other phases or applications. How can the brittleness of these secret-keeping schemes be compared?

The following paragraphs introduce a simplified system metric for brittleness due to secret-keeping in authentication systems, then analyzes certain deficiencies of this metric and proposes certain augmentations to cope with those deficiencies.

One simplified approach to a system metric for the brittleness of authentication due to the fragility of secret-keeping mechanisms is to use a binary weighted assessment of reliance on secret-keeping throughout each portion of the life cycle. The earlier that the dependence on secret-keeping appears in the life cycle, the greater the brittleness, since there is more time for such secret-keeping to be subverted.

In Figure 5, a binary value of "1" or "0" is recorded for each phase of the system life cycle, depending on whether authentication depends on any "secret-keeping" in that portion of the life cycle. This is called the "binary secret-keeping vector." Protection mechanisms themselves that are kept secret (rather than simply cryptovariables of a mechanism that is not secret) cause greater brittleness and show up in the "creation" portion of the life cycle and are weighted more heavily since there is more time for these secrets to become subverted or extracted. In the notional example depicted in figure, the brittleness of secret keeping in the depicted authentication system is the decimal value of the binary vector, or 79. An authentication system

that depended on keeping secrets in every one of the nine[41] segments of the depicted life cycle would have a brittleness of $(2^9 - 1)$ or 511. An authentication system that depended on no secrets at all in any phase of the life cycle would have brittleness zero.

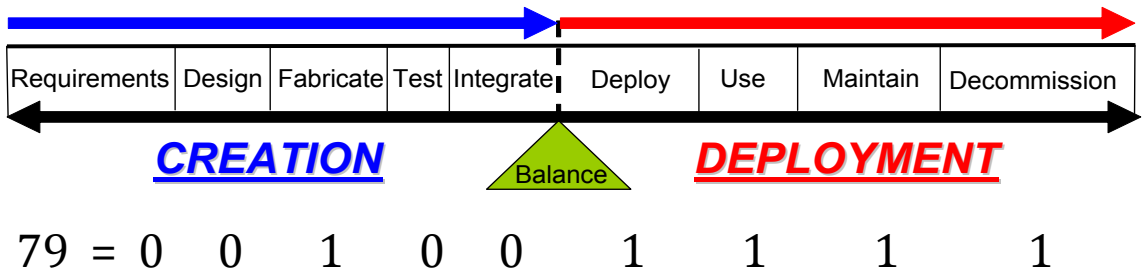

Figure 5. Brittleness of Authentication due to Secret-Keeping.

This simplified metric achieves some of the objectives of analyzing brittleness characteristics over the entire life cycle, but fails to account for different exposure of secrets in different application scenarios. For example, a secret established and maintained in a locked or guarded facility should be less brittle than a secret carried around on one's person in the form of a smart card. The simplified metric also does not attempt to measure the effectiveness of "anti-reverse-engineering barriers" (protection mechanisms kept secret). This is justified for high exposure environments because (as experts interviewed in the course of this project agree) these "anti-reverse-engineering barriers" are intended only to increase the cost of and to delay the adversary's secret extraction. These "anti-reverse-engineering barriers" add value only in low exposure environments in which the adversary has little or no access to a sufficient number of units for reverse engineering, but add little or no value in a high exposure environment where a motivated adversary can bring large resources to bear to breach the protection mechanism.

In order to incorporate the concept of "exposure to the adversary" into this scheme, we define "exposure to the adversary" as the $\log_{10}$ of the number of people able to access to the components of the authentication system that contain secrets in any given portion of the life cycle. We further separate the concept of exposure into "design exposure" and "operational exposure". Design exposure pertains to access to information regarding the requirements, design, or "as-built" data. Operational exposure pertains to access to operational variables only available in specific operating components. We note that design exposure is a metric pertinent to any phase of the life cycle (the design information may be extracted from the earliest "requirements" or "design" phase or from the reverse engineering of components accessed in the "decommission" phase). Operational exposure is a metric that attempts to characterize exposure of operational variables that are typically present only during portions of the "deploy", "use", "maintain", or "decommission" portions of the life cycle. Design exposure does not imply operational exposure, but to exploit information gained from operational exposure, an adversary

[41] The division of life cycle into nine parts is rather arbitrary. A decomposition of greater or lesser granularity may be appropriate.

must typically achieve understanding of the design (design exposure). We also note that Design exposure pertains to the ability of an adversary to extract design/as-built information, not to modify or insert design information, which would require a similar but separate metric we might call "subversion exposure" for each phase of the life cycle.

Figure 6 depicts an example of the "fabrication" phase of the life cycle. If the "fabrication" phase is restricted, permitting access by only 10 people, then the design exposure during the "fabrication" phase is 1. If the deployment phase depends on a secret kept in a smart card, and the smart card is a commercial unit that can be acquired by anyone on the planet, then the design exposure[42] of the deployment phase is $\log^{10}(10^9) = 9$ (since any person or adversary is able to acquire sufficient units to successfully reverse engineer, given the time and resources available to a sophisticated adversary). For the example depicted in Figure 6, we presume that the other phases of the creation portion of the life cycle are exposed to fewer than 5 people, so the design exposure of these phases is log10 (~1) = 0, and that wide exposure to reverse engineering would persist over the entire deployment portion of the life cycle so the design exposure over these phases would remain 9.

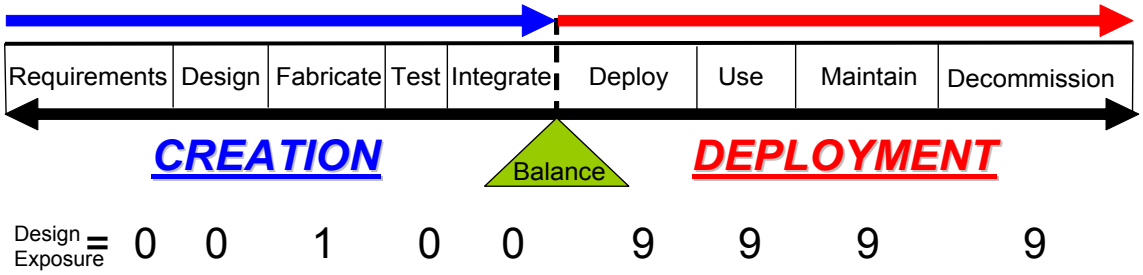

Figure 6. Design Exposure Vector Example.

Figure 7 depicts a notional example of "operational exposure". In this example, there are no operational variables to be exposed in the creation portion of the life cycle so the operational exposure in this part of the life cycle is zero (in theory, things such as mask-selected secret options in an otherwise open design that are invoked in the fabrication phase could be considered operational variables and could make the operational exposure of these phases non-zero). For this example, we assess that operational cryptovariables are only instantiated after deployment and prior to "use", but are destroyed before "maintenance" or "decommission". In this notional scenario, only the operational exposure of the "use" phase is non-zero. If the number of people who have easy access to a keyed-up smart card in an individual's possession is approximately 100 (close family members and friends, etc. who might find the card temporarily unattended over the course of the "use" portion of the life cycle), then the operational exposure of the "use" phase of the life cycle is 2.

[42] Assuming a world population of over 6.79×10^9 and arbitrarily rounding down to 10^9 (for the simplicity of working with single digits 0 through 9.

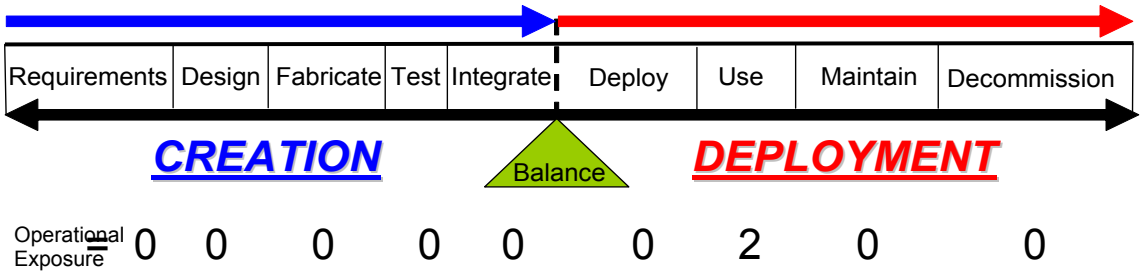

Figure 7. Operational Exposure Vector Example.

In the course of considering hypothetical authentication systems using these metrics, we observe that a multitude of metric vectors are required. For example, the binary "secret-keeping" vector described is useful to decompose into a separate vector for design secrets and one for operational secrets (cryptovariables). These separate vectors can then be combined with the separate exposure vectors described above. These vectors can eventually be converted into scalar numbers for raw comparisons, but at this stage of development, it is more useful to manipulate the entire vectors that correspond to assessments of brittleness and exposure in different portions of the life cycle.

The limits of the resources of this feasibility study project have not enabled full development of this approach to brittleness metrics, but it appears promising. The next step would involve application of these metrics to describe real authentication systems and feedback of the deficiencies into refinements of the metric, assisted by experts in existing authentication systems.

CONCLUSION

Our national cyber infrastructure exists in a high exposure environment[43] and is vulnerable to multiple adversaries as evidenced by daily cyber attacks reported in the public literature. This work examined feasibility of an authentication concept that could greatly improve our ability to apply attribution and deterrence concepts to protection of DOE and other national cyber infrastructure (which has proven to be extremely difficult in high exposure environments without improved authentication tools).

Secret-keeping in Authentication systems leads to brittleness in the sense that extraction or discovery of these secrets by an adversary leads to catastrophic failure of the authentication system, because the adversary then has sufficient information to spoof the authentication.

Authentication systems that do not rely on secret-keeping would be immune to this kind of brittleness. Authentication systems that do not use secrets are possible, but appear to be difficult to implement in terms of data volume that must be processed. Research into specific methods of authenticating components, messages, and individuals without relying on secrets will likely result in far more resilient authentication systems than we have today. Specifically, research into the use of unique and hard-to-clone characteristics of devices are expected to have high payoff. In particular, investigation of methods that leverage the use of Physically Unclonable Functions or PUFs to generate easily identifiable characteristics should be high priority. While the measurement of these PUFs can be regarded as a hard-to-extract secret, with proper design the secret may be made less brittle than conventional authentication secrets generated and held by other means.

Since most of the authentication systems in use today involve secret-keeping in one or more phases of the life cycle, metrics of this brittleness will assist in the analysis of the resilience of these systems.

This report has outlined an approach to developing metrics for authentication system brittleness that looks promising, but needs refinement and improvement by application to analysis of real systems. We believe that metrics that identify which portions of the life cycle rely on secret-keeping, augmented with metrics for the exposure of authentication components to the adversary during each phase of the life cycle will enable more robust analysis of these systems, and will enable comparison of resilience of different authentication schemes against these kinds of failures.

This study recommends future research into authentication systems that do not depend on secrets. Such research should be well-grounded in knowledge of current authentication systems and PUF techniques, and have relatively concrete ideas regarding how to differentiate a real-time measurement of a hard-to-clone characteristic from a playback (also without resorting to secret-keeping). These techniques show great promise for chip verification (assurance against wholesale substitution or counterfeiting), Chip-to-chip authentication (for component binding

[43] Over 85% of the U.S Critical Cyber Infrastructure is privately owned, as described in the DHS' National Assets Database and in "The National Strategy for Homeland Security: Office of Homeland Security," 16 July 2002, available at http://www.whitehouse.gov/homeland/book/nat_strat_hls.pdf.

and configuration control in high assurance systems), as well as for authentication of messages, network transactions, and individuals.

Distribution

Electronic Copies

1	MS0352	G. R. Anderson	1718
1	MS1084	T. M. Bauer	1746
1	MS0341	P. J. Robertson	1751
1	MS0341	R. Lovejoy	1751
1	MS1072	D. V. Campbell	1767
1	MS0868	T. E. Zipperian	2700
1	MS0671	J. R. Hamlet	5627
1	MS0780	E. L. Witzke	6525
1	MS0769	R. L. Hutchinson	8970
1	MS0806	J. H. Naegle	9336
1	MS0359	D. Chavez, LDRD Office	1911
1	MS9018	Central Technical Files	8944
1	MS0899	Technical Library	4536


Sandia National Laboratories