proof space

ProofMark™
System Technical Overview

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

proofspace

Table of Contents

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

The ProofSpace system is designed to verify to a high degree of trustworthiness the "time existence of data." To accomplish this, it employs a patented transient key technology to irrefutably link a given set of data (a digital file, for example) to a given interval of time. The linkage is embodied in a ProofMark, which is a certificate containing multiple cryptographic mechanisms that attest to the existence of the original data within the stated time interval.

The ProofSpace solution is most easily understood by beginning with a description of the fundamental conceptual building blocks of the system. We will then provide a functional description of the current ProofSpace implementation, followed by a hypothetical scenario illustrating how ProofMarks might be used with an E-Mail Archiving/eDiscovery solution, and conclude with a discussion of alignment with relevant industry standards for time and date stamping and file authentication.

Fundamental Concepts

The foundation of the ProofSpace system is transient key technology, which is essentially asymmetric key cryptography with a twist: private signing keys are created and destroyed at regular time intervals. This provides two intrinsic properties:

1. Binding of a private cryptographic signing key to an interval of time.
2. Private keys do not need to be protected as in classic public key systems (e.g. PKI), permitting widespread cross-certification and publication of verification information.

Public Key Cryptography

Public key cryptography (often also termed asymmetric key cryptography) is based on natural asymmetries in the ability to perform certain mathematic computations[1]. In the 1970s, algorithms were published that leveraged these mathematical principles to provide (among other things) the ability to sign digital data, establishing a mathematically irrefutable endorsement or authorization, as well as a tamper-evident integrity "seal," on the data.

Public key cryptography has been studied rigorously, and implemented widely, with an impeccable track record. It now forms the underpinnings of modern security systems ranging from military applications to the global banking system. Modern asymmetric cryptography signatures are considered computationally unassailable using current technologies.

However, by design, public key cryptography remains vulnerable to inappropriate disclosure of the signing key (more popularly termed the private key to denote the requirement for maintaining its secrecy[2]). Anyone with possession of the private key can forge signatures, authorized or not.

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

1 "Public-key cryptography." http://en.wikipedia.org/wiki/Public-key_cryptography

2 The private key is also meant to be kept secret because it is used to decrypt confidential data, but since the ProofMark system is not concerned with encryption, we do not consider that here.

proof space

Transient Keys: A Solution to Signing Key Vulnerability

The primary innovation of the ProofSpace system is to eliminate the risk conventionally associated with a private signing key. This is accomplished through the combination of making the signing key transient and creating a digest log of the requests received during an interval. The digest log is itself digested and signed by one or more transient keys. This addresses the need to maintain the secrecy of private cryptographic signing keys by simply eliminating the existence and utility of the signing key, using three mechanisms:

1. ProofSpace does not issue keys to humans (who are prone to misplacing or misuse things), but rather to time. Or, more accurately, an interval of time, such as "between 9:00 and 9:05 AM, UTC"
2. All requests to the system are accumulated in a sequential concatenation
3. At the end of the time interval, the private key is destroyed, preventing it from ever being disclosed in the future.

As noted earlier, this provides several intrinsic and practically useful properties:

1. The private signing key is associated with the interval of time in which it exists. Thus, any data that is signed by this key is by extension proved to have existed by at least that time interval.
2. All signing events of the private key are recorded in a sequential log established during the interval.
3. Data cannot be fraudulently signed after the fact (post-dated) since the private key is destroyed at the end of the interval (and thus cannot physically be used to sign further data).
4. If by any means a private key were to be re-created (future computational advances) or hypothetically not destroyed, it would pose no risk because it would be impossible to falsely insert a signing event into the log created during a previous interval.
5. Verification of the signature can be performed readily using the public key, which, by the original design of asymmetric key cryptosystems, is intended to be published to wide audiences. It can even be persisted with the signed data itself.
6. There is no need to provide extensive infrastructure to protect private keys, since they are neither persisted nor usable after the conclusion of an interval (in contrast with other asymmetric cryptosystems like Public Key Infrastructures, or PKI).

Transient key signatures timestamp and authenticate a set of data by cryptographically signing it using a transient key associated with an interval of time. Combined with certain other mechanisms (discussed in more detail below), this provides a high-fidelity time and date stamping and file authentication solution.

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

proofspace

Functional Overview

The ProofMark system is comprised of the following components:

The ProofMark Server. A network server/service that:

a. Bootstraps and maintains the infrastructure (Interval Chains) necessary to issue ProofMarks.

b. Issues and Verifies ProofMarks.

c. Maintains Archives of issued ProofMarks and Interval Chains.

d. Cross-certifies ProofMark infrastructure on other servers.

The ProofMark Client. An XML-based client API (easily integrated within existing applications or services) that:

a. Requests ProofMarks.

b. Verifies ProofMarks (either through self-verification, or against a remote Archive).

Figure 1 presents an overview of how these components work together. In this illustration, a user application completes a transaction and requests a ProofMark accompanied receipt (1), the Application Server requests a ProofMark via the ProofMark Client, which is received by the ProofMark Server (2), which then generates a ProofMark and provides it to the user (3). Later, the user's local ProofMark Client verifies the ProofMark at the original server (4). In addition, a second ProofMark Server is shown cross-certifying the Interval Chain on the first.

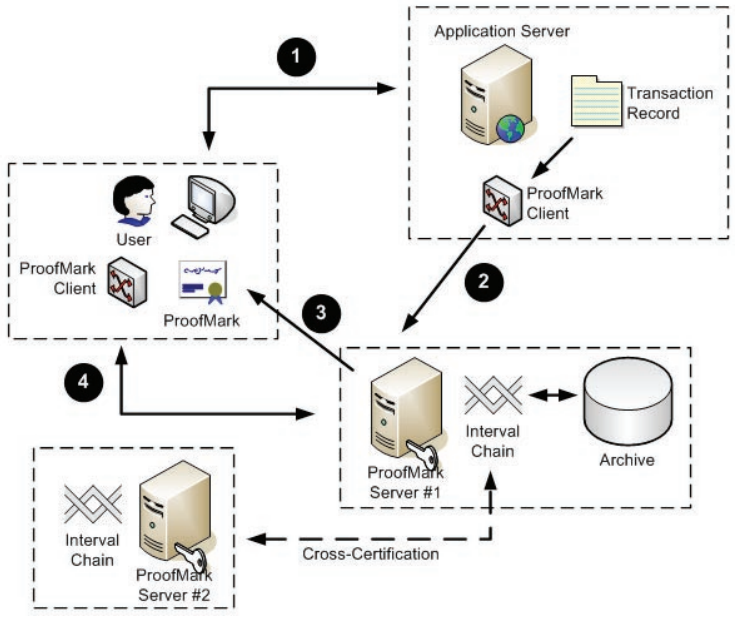

Figure 1: The primary elements of the ProofSpace system. Numbered steps are discussed in the main text.

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

Each component is described in further detail and related to the overall function of the system below.

The ProofMark Server

The primary functions of the ProofMark Server are to issue and verify ProofMark Certificates, which are the certificates of time and authenticity. To accomplish this, it performs the following activities:

1. Obtains time from a trustworthy source.
2. Generates Intervals and their associated transient signing keys
3. Manages Chains of continuous Intervals
4. Issues ProofMarks in response to client requests
5. Maintains a rolling Digest Log of ProofMarks issued across each interval.
6. Archives and publishes Interval Chains to service ProofMark verification requests.
7. Verifies previously issued ProofMarks.
8. Cross-Certify Interval Chains on other servers

Each of these activities is described in more detail below.

Trustworthy Time

Each ProofMark has a timestamp indicating the time that the ProofMark was issued. The timestamp is created using Universal Coordinated Time (UTC), with precision to the nearest millisecond. Within a ProofMark Server, timestamps are obtained from a trusted Time Source (commonly via the Network Time Protocol, NTP, although more robust mechanisms are supported). Times are calculated via a time biasing mechanism, which obtains the time from the trusted time source periodically and uses a local hardware timer in the interim. If the trusted time cannot be obtained, the ProofMark server will not issue ProofMarks until the trusted time can be reestablished. The system clock, which is vulnerable to tampering, is never used as a source of time.

Every timestamp has an associated accuracy, in milliseconds, which is reported along with the timestamp in every issued ProofMark. In a typical configuration, accuracy within 100 milliseconds of the Atomic clock is possible. If the Time Source is not running within its specified tolerance, a Stale Time Exception occurs, which prevents the creation of ProofMarks.

Intervals

Intervals are used by the ProofMark system to provide the transient key pairs that signs the data in a ProofMark. As noted earlier, using transient key pairs greatly reduces the exposure of the private key to theft or compromise.

The length of time during which a key-pair can be used is set during start-up of an issuing ProofMark server. Each server generates one key-pair per Interval.

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

proof space

A single ProofMark server has only one active Interval at any given time. As the server runs, subsequent Intervals are created which are guaranteed to be contiguous (the stop time of an Interval is identical to the start time of the next Interval). These contiguous Intervals form a chain of keys, with each Interval's public key being signed by the previous Interval's private key.

At the end of each Interval the corresponding private key is destroyed and a new key pair is generated for the subsequent Interval. During the process of activating a new Interval, the current Interval's private key signs the new Interval's public key and start and stop times. Once a signature for the Interval's key has been acquired, the private key is permanently destroyed.

Figure 2 illustrates this process.

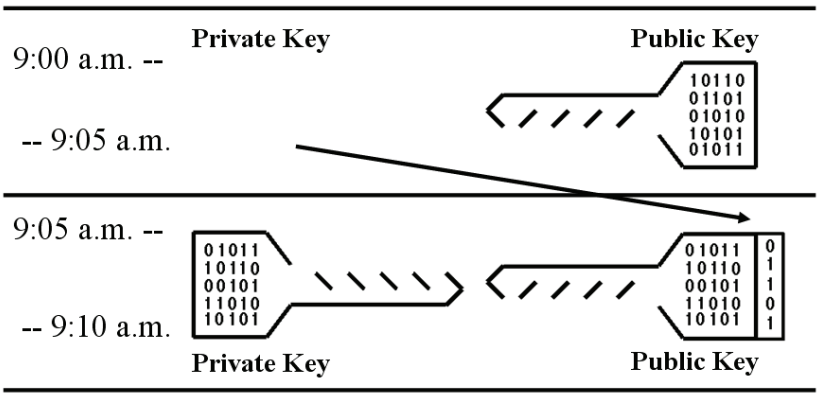

Figure 2: The prior Interval private key is used to sign the new Interval public key. The prior Interval private key is then destroyed.

If a new Interval cannot be readied and prepared before its prescribed start time, the chain is broken, and the server automatically restarts a new chain.

An Interval contains the following information:

- The server-id (the hostname[:port] of the server)
- The start time of the Interval chain in UTC
- The start time of the Interval in UTC
- The stop time of the Interval in UTC
- The public key for the Interval
- The digital signature of the Interval's public key, signed by the previous Interval's private key
- The X.509 digital certificate for the Interval, issued by the ProofSpace server
- Cross-certification information (a ProofMark issued for an Interval by another ProofMark server, see Cross-Certification)
- A hash of the previous interval's entire digest log
- A digest log, or "stack" of hashes from every incoming ProofMark request during the interval so far.

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

proof space

Interval Chains

The ProofMark Server creates Intervals contiguously in a chain that continues to lengthen with the passage of time (rather like a stream of ticker tape issued by stock ticker machines invented in the late 1800's).

The start time of the very first Interval in the chain is known as the chain start time, and is stored in each Interval. While theoretically possible, it is unlikely that two different servers would be configured with the same server-id (the hostname[:port] of the server). It is highly improbable that these servers could also be started at exactly the same time, resulting in identical chain start-times. Therefore, adding the chain start time to the server-id uniquely identifies an Interval chain. Once the chain is identified, an Interval within the chain is uniquely identified by the Interval's start time. The chain's Intervals are stored persistently in an archive (see Archive, below).

Using Intervals to Issue ProofMarks

During each Interval, the private key is used in the creation of ProofMarks, which are the timestamping and authenticity certificates.[3] Many ProofMarks can be issued during an Interval, each signed by the Interval's private key.

Certifying data using a ProofMark is very straightforward: the submitted data is simply signed by the Interval private key. The original data, the signature, and the Interval public key are all included in the ProofMark. One can now simply use published cryptographic routines to verify that public key applied to the signature resolves to the original data, illustrating the self-contained integrity "seal" provided by a ProofMark.

The start time within each Interval coupled with the chain start time form an unbroken sequence of public keys that can be used to fix a ProofMarks position in time, which also fixes the exact state of a set of data at that point in time. To prove this state at some future point, the chain of public keys is posted to an easily accessible place (i.e. several web servers) from where they can be used to verify a ProofMark (see Verification below).

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

3 The technical specifications for ProofMarks, including a document type definition, are available upon request.

proof space

Digest Log

To further protect against fraudulent pre- or post-dating of ProofMarks, the ProofSpace Server creates a concatenate digest[4] log of all ProofMark requests and records it within the corresponding Interval. At the end of each interval, a digest is created from the digest log itself, and then inserted into the new interval, prior to the signature by the prior private key. This is illustrated in Figure 3.

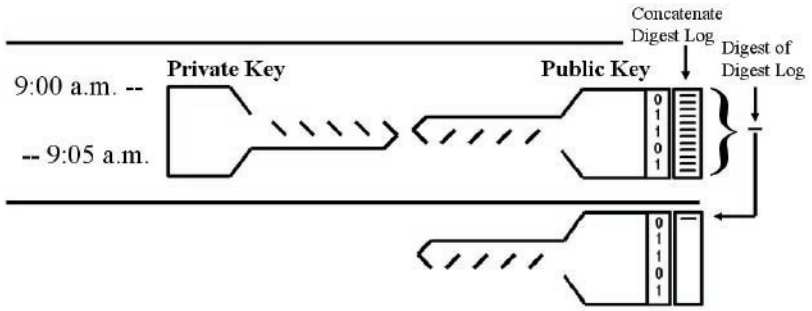

Figure 3: Use of a rolling digest log to prevent fraudulent post facto injection of ProofMarks into an Interval Chain.

Since the sequence and content of ProofMark requests is unlikely to be known in advance, this feature constrains generation of fraudulently pre- and post-dated ProofMarks.

Cross-Certification

To provide the most robust protection against the compromise of a single chain, Interval Chains may be cross- certified with other ProofMark Server instances. Cross-certification refers to the process by which one ProofMark server issues a ProofMark for another ProofMark server's Interval. The actual Cross-certifications are ProofMarks whose signed data is an Interval. Effectively, the Interval private keys from the Cross- certifying Server are used to sign the Interval public keys on the certified Server. Figure 4 illustrates two ProofMark Server Intervals Cross-certifying an Interval from a third.

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

4 A digest, or hash, is a small digital "fingerprint" of a given set of data produced by algorithms that are commonly used to validate digital authenticity and integrity [reference].

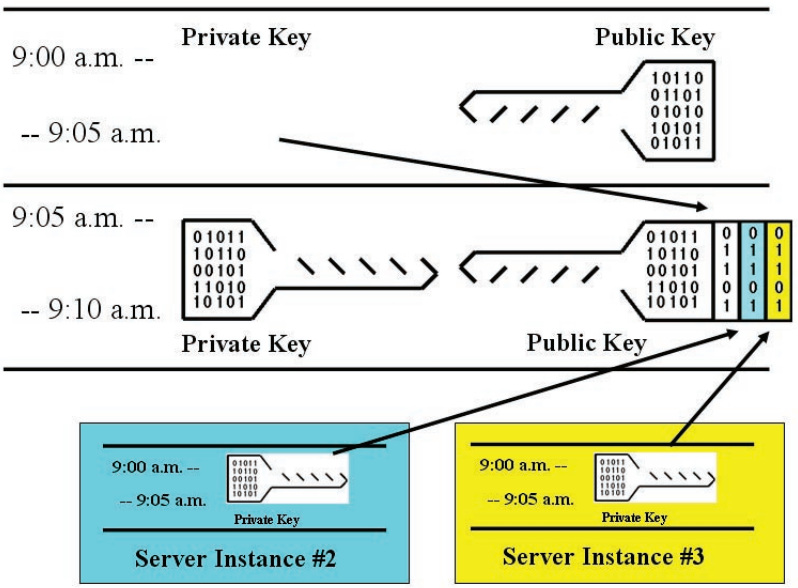

Figure 4: ProofMark Server Instances #2 and #3 Cross-certify an Interval generated by a third Server.

The Cross-certification process requires that the timestamp (from a trusted time source) of the Interval and the timestamp of the Cross-certifying server agree. That means the difference is less than the sum of the accuracies of the two timestamps plus the time required to obtain the Cross-certification.

Cross-certification can optionally be configured such that cross-certifying servers authenticate each other using additional methods, such as classic PKI certificates.

Cross-certification provides independent proof of the existence of the Interval (and its public key) at a point in time. If independently operated Servers are used for Cross-certification, this creates a widely witnessed chain of proof for the Interval, greatly enhancing the trustworthiness of ProofMarks issued. It also creates a virtual web of proof across a distributed network, as illustrated in Figure 5.

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

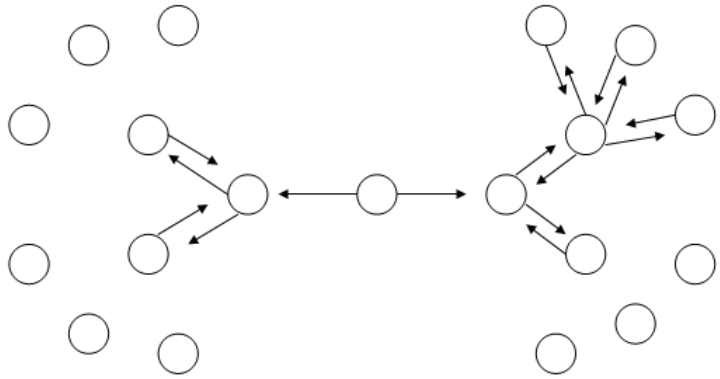

Figure 5: Cross-certifying ProofMark Servers create a virtual web of proof across distributed systems.

In contrast to classic Public Key Infrastructure (PKI) architectures, wherein compromise of a single certificate authority (CA) can have devastating effects on overall trustworthiness, the ProofSpace system remains as "strong as its strongest link." ProofMarks issued by a ProofMark System are as trustworthy as those issued by the most reliable of the Cross-certifying Servers. This is exactly converse to PKI which is considered as "weak as its weakest link". PKI certificates are considered only as reliable as the "weakest" cross-certifying authority.

An Interval can have any number of Cross-certifications, issued either by other servers within the same organization, or by servers in other organizations. ProofMark Servers can be configured to require a minimum number of Cross- certifications before an Interval can become active. A larger number of Cross-certifications results in a more widely witnessed chain of proof.

Cross-certifications can also authenticate another server's time. Cross-certification also protects the archive from tampering, since the Cross-certification web extends to several archives and replicas of those archives. To falsely use the private key of an interval would require access to and control of the entire web of cross-certifying archives.

Archives

An Archive is a database in which Intervals and their Cross-certifications are stored. The ability to retrieve an Interval and its Cross-certifications from an archive provides all the information necessary to complete the verification of a ProofMark.

Because an archive is a logical database, it can be shared or replicated (copied) to many servers, and can be hosted on any server. Its physical persistence may be mapped into either a normal file system or a JDBC-compliant (Java Database Connectivity) relational database.

A unique hostname URL:hostname or hostname:port identifies each archive, where the port defaults to 80. This host name is the logical host of the archive, which may be either a single real server or a load-balance proxy to a group of servers. Other hosts may have replicas of the archive as well. This is the service that is contacted fro Verification (see below).

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

If the archive's real host ceases to exist, the ProofMark archive Directory (see below) will list forwarding host addresses where copies of the archive are located.

Every Interval must be stored in at least one Archive, known as the Interval's root archive. Intervals may be stored in additional archives as well. During creation of the Interval, an archive Tree is established for the Interval and the Interval is stored or published in its root archive before it is available for use. After its initial publication, the Interval is forwarded asynchronously to one or more additional archives in the archive Tree, which may in turn each forward to additional archives. The archive Tree is represented as part of the Interval's XML representation and therefore appears in each ProofMark issued by the Interval. This enables the holder of the ProofMark to know which Archives can be used for later verification of the ProofMark. In a typical situation, an organization will have its own archive, and will forward its Intervals to a public archive, but more extensive archive Trees are possible. Each additional archive may have been configured to forward to another level of archive (propagating the archives).

The process of establishing the archive Tree for an Interval occurs immediately after the Cross-certifications for the Interval have been obtained. The archive Tree is constructed by combining the archive Trees from the servers that issued Cross-certifications as follows:

- The Interval's local archive becomes the root of the archive Tree
- The set of archive trees of all of the Cross-certifications for the Interval are added as immediate branches of the root archive
- If there are archives that have been configured for publication, without requiring Cross-certifications, these archives are also added as branches
- Any cycles or redundant branches in the resulting archive tree are removed In an exception to this process, the Interval does not have a local archive. In this case, it must be configured with only a single Cross-certification group from which Cross-certifications are required. The resulting archive Tree then becomes a copy of the archive Tree from that group. See the Load Balancing Topology in the Planning section below for an example.

Publication

Publication refers to the process of making Intervals and their Cross-certifications available in one or more databases that are:

- Permanently accessible, even if the issuing organization ceases to exist
- Stored in such a way that they cannot be altered without detection

Publication is achieved in the ProofMark system with the following processes:

- An Interval and its Cross-certifications are published to the root archive in the Interval's archive Tree, before the Interval can become active
- An archive can be periodically replicated to several servers in order to provide high availability and redundancy against loss

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

- Intervals and their Cross-certifications are propagated from one archive to another, as defined by the subordinate branches of the Intervals archive Tree, using the following automatic process:
  - As an Interval is stored in any archive, it is flagged for propagation if there are any branches in the Interval's archive Tree that occur beneath the archive in which the Interval is currently being stored
  - Periodically, a ProofMark propagation service forwards all Intervals marked in this way to each of the archives that appear beneath the current archive in the Interval's archive Tree (the propagation flag for the Interval is cleared when the Interval has been propagated successfully to each of these archives)
  - This recursive process continues until the Interval has eventually been stored in each archive in its archive Tree

Replication is important, even for verification-only servers, since Interval publishing and propagation only distribute Interval information to a single server in each archive group.

Verification

Once a ProofMark is issued, it is straightforward to determine that it has not been tampered with and that it is authentic.

To determine that a ProofMark has not been tampered with subsequent to issuance, a simple internal consistency check can be performed: the signature can be validated against the public key. This provides for a simple, on-the-spot verification of integrity using widely available software cryptographic routines.[5] Communication with a ProofMark Server is not required.

To confirm a ProofMarks authenticity, it must be verified against an Archive. There are several levels of Archive verification. All levels of Archive verification perform the internal verification described above prior to checking the archive. Each adds an additional level of authentication, preventing against increasingly sophisticated attacks. Each level also takes additional computing resources to complete.

The levels of Archive verification are listed below, in order of ascending strength:

- ProofMark internal verification
- Interval verification (the first level of archive verification)
- Cross-certification verification (the second level of archive verification)
- Digest log verification (the third level of archive verification)

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

5 One may also compare a digest (or hash) of the original data with the digest included in the ProofMark to validate that the signed data itself was not altered. This can also be performed in the absence of communication to a ProofMark Server using a simple software comparative operation.

The relative strengths of each Verification level are illustrated in Figure 6 (including internal verification). Each Archive Verification level is describe in more detail below.

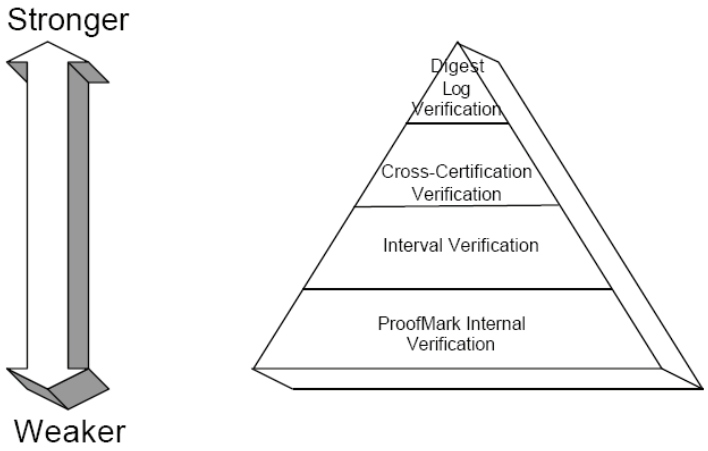

Figure 6: Relative Verification level strengths.

Interval verification

The first level of archive verification authenticates any PKI signatures that were included in the original request that generated the ProofMark (these are part of the ProofMark). Authentication is accomplished by first verifying each certificate in the PKI signature's certificate chain, then checking for a trusted certificate in the machine's local keystore whose subjectDN matches the issuerDN of the first certificate in the PKI signature's certificate chain. If these keys fail to match, an error is reported in the verification report.

Cross-certification verification

The second level of archive verification authenticates the PKI signatures, and checks the archive for the public key of the Interval. Then, the Interval's Cross-certifications (which are themselves ProofMarks) existing in the Archive are recursively authenticated.

Digest log verification

The highest level of archive verification authenticates the PKI signatures, checks the archive for the public key of the Interval, and checks the Interval's Cross-certifications. When these have been verified, the server confirms that the ProofMark digest exists in the Interval's archived digest log. ProofMark verification reports The ProofMark server issues a ProofMark verification report in response to a verification request. Input to this request is the ProofMark certification (the XML) to be verified. Output from this request is a Verification Report XML document containing the results.[6] The Verification Report either lists any errors discovered in the process or indicates that the verification was successful.

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

6 The technical specifications for a Verification Report, including a document type definition, are available upon request.

proof space

Internal security

The ProofMark system, by design, remains truly resilient in the face of compromise, whether malicious or inadvertent. As noted in the technical overview, because of its use of transient key technology, there is really nothing to compromise from a confidentiality perspective. From an integrity perspective, the implementation of a rolling digest log, combined with independent cross-certification, makes the ProofMark system highly secure and has been described as the first commercial example of a forward-secure system.

The ProofMark Client

The ProofMark Client is a HTTP-based application programming interface (API) that is easily embedded within existing applications or services. It provides two primary functions:

- Requests ProofMarks.
- Verifies ProofMarks.

Each of these functions is described in more detail below.

Standards Alignment

ProofSpace believes strongly in industry standards as a mechanism for driving the interoperability necessary for a functional global ecosystem, and in fact was instrumental in drafting the X9.95 Trusted Time Stamp Management and Security Standard.

X9.95 Trusted Time Stamp Management and Security Standard

As noted above, ProofSpace was instrumental in drafting X9.95 as a member of the Accredited Standards Committee X9. In fact, ProofSpace is one of the primary authors of the evolving X9.95 application programming interface (API).

The X9.95 standard includes multiple methods for securely generating trusted time stamp tokens. X9.95 includes the basic logic and syntax for what can be developed as a Universal Client API. A Universal Client API is a small element of code that can be incorporated into business and consumer applications, and generates and/or manages data appropriate for trusted time stamping. A Universal Client API can request a trusted time stamp token from any of the X9.95 standards-compliant methods and verify a trusted time stamp token from any of the X9.95 standards-compliant methods.

For an example of how this would work, let us consider a business or consumer application (say a document authoring application such as a word processor, a spreadsheet, or even e-mail). The business/consumer application would invoke (make a program call) for a trusted time stamp for a given set of data at the earliest instantiation of the data...say when it is first saved in the case of a word processor or spreadsheet or as it is sent in the case of e-mail. The business or consumer application would then interact with the Universal Client API which would then handle all the work of generating a properly formatted trusted time stamp token request, forwarding this request to the designated source of the trusted time stamp token (as designated by the end user organization), assuring a request acknowledgement and then completing the process of receiving the trusted time stamp token from the generating method/system and returning the trusted time stamp token to the business or consumer application which authored the data.

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

What this means to end-user organizations or individuals who generate or manage data which for which a trusted time stamp token would be beneficial is they will have the ability to establish a business relationship with a service provider or obtain for themselves and operate a product which can generate an X9.95- compliant Trusted Time Stamp tokens. The end user organization would not be locked into any one provider or product. They would preserve the ability to make a procurement decision based on the best available solution for their needs from among competitively available sources...and they would further maintain the ability to change their choice without having to change their business/consumer application as circumstances shift (including evolving end-user needs, market factors such as pricing, regulatory policy moves or other competitive considerations).

What this means for end-user organizations who encounter data accompanied by an X9.95-compliant trusted time stamp token is the ability to verify it regardless of which of the X9.95-compliant methods had been used to generate the trusted time stamp token. Examples of such entities, more commonly referred to as relying parties, might include but are not limited to trading partners, auditors, regulators, etc. This ability to verify the X9.95-compliant trusted time stamp token would not require the relying party to have its own direct commercial/contractual relationship with the originating application or Trusted Time Stamp token vendor. This feature is of X9.95-compliant solutions is of critical importance, imparting essential value on data because it provides for the portability of data together with the associated trusted time stamp token across custodial and control boundaries.

What this means for application architects and developers is they can directly incorporate X9.95-based trusted time stamping into their application, enabling the ability to request and/or verify an X9.95- compliant trusted time stamp token without having to commit to any one particular method or vendor. In this way, architects and developers can defer on the ultimate choice of which of the X9.95-compliant methods to use and thereby enable the end-users of their applications to make the choice which is best suited to their needs.

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

proof space

ProofMark Requests

A ProofMark request contains the following information:

- A reference to the data associated with the ProofMark, such as a filename or a SQL string or the actual data to be certified (optionally used based on business circumstances such as when the amount of data to be certified is relatively small and can be included in the request)
- An SHA-based (SHA-256, -512, etc.) digest of the contents of the data or the data referred to by the reference (the digest is prepared by your client program when creating the ProofMark Request)
- Zero or more X.509 certificates acting as witnesses to the request (to include X.509 certificates, your application must provide the signed hash of the transaction data to the ProofMark client API)

There are additional options that can be used when requesting a ProofMark, indicating whether the ProofMark should be stored on the ProofMark server or whether only a reference to the certificate should be returned.

Verification Requests

The Client API also supports verification of previously issued ProofMarks.

There are several levels of verification (as noted earlier, reiterated here):

- An internal consistency check (validating the signature within the ProofMark Certificate using the public key)
- Sending the ProofMark to a ProofMark archive server for authentication
- Recursively verifying the integrity of the Interval chain using Cross-certifications
- Recursively verifying the integrity of the Interval chain using Cross-certifications and checking the digest log for the digest of the ProofMark being verified

Each level except the internal consistency check produces an XML verification report. If the ProofMark has been tampered with, the report will indicate what errors were uncovered. Using the more thorough levels of verification impacts the amount of CPU time required to complete the verification.

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

proof space

Example Scenario: ProofMarked Email Archiving & eDiscovery

The following scenario illustrates how the ProofSpace system is typically applied in Email Archiving applications.

1. Joe Smith, the network administrator for Amalgamated Industries, uses Symantec Enterprise Vault to automatically archive email and other documents on the network. Joe has also installed the ProofSpace product, ProofMark for Symantec Enterprise Vault, which adds ProofMarking features to the system.
2. As records and emails are ingested into the Enterprise Vault application, they are automatically ProofMarked to include a trusted timestamp and provable data integrity seal.
3. 18 months later, Amalgamated is involved in a lawsuit, and opposing counsel demands the email archives as part of the discovery phase of the trial.
4. Amalgamated draws the judge's attention to several emails which prove that Amalgamated is innocent of the charges brought against it. Opposing counsel challenges the authenticity of those emails, claiming that Amalgamated has planted them (post facto) into the archive in order to make itself "look better" to the court.
5. Amalgamated presents the ProofMarks associated with the emails in question to the Court, and proves that they could not have tampered with the archive, or inserted false records the archive without invalidating the ProofMarks. Amalgamated wins the case.

Figure 7 illustrates this scenario, with relevant steps highlighted.

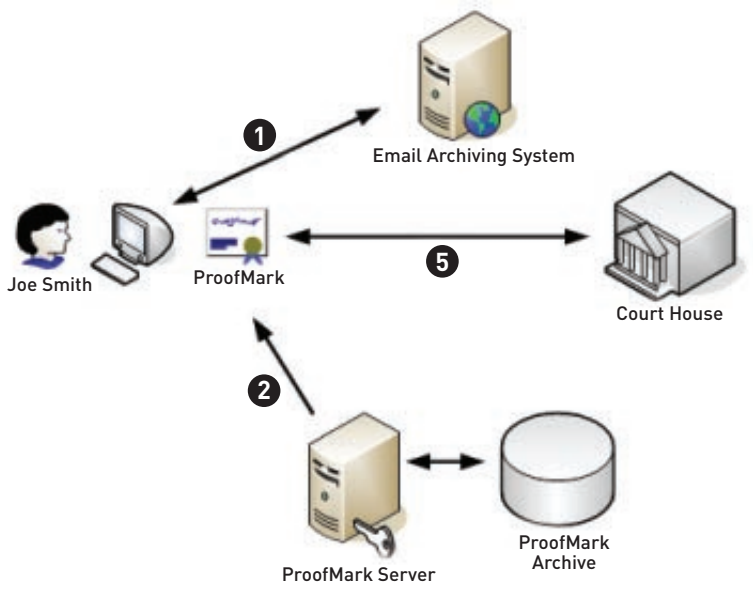

Figure 7: A hypothetical e-mail archiving scenario. Numbers refer to relevant transactions described in main text.

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823
www.proofspace.com

Conclusion

Transient key technology provides the foundation of the ProofMark, a form of trusted time stamp that authenticates a set of data by cryptographically signing it using private asymmetric keys that exist only within discrete intervals of time. Combined with certain other mechanisms including digest logging of ProofMark requests, cross-certification, and publication of verification information, this provides a "widely witnessed web of trust" that could easily form the basis of a high-fidelity, standards-compliant time/date stamping and file authentication system.

ProofSpace
900 Clancy Ave NE
Grand Rapids, MI 49503
(312) 933.8823