Secure…
Cryptographic Time-Stamps
Through Transient Key Technology

By: Paul Doyle of ProofSpace
We Will Describe A System That…

• Is a method of self-validating proof of time
• Creates Cryptographic Timestamps that never expire
• Is a fully distributed system
• Is immune from the compromise of secret keys
• Is independent of a Trusted Third Party
• Creates a network of validation & verification
The ProofMark System™ Can Be Used To Prove Integrity And Time Existence…

Original Data

…010001100…

…For Any Set Of Data

…Or Any Record
Regardless Of The Application…We Can Enable You To Create And Maintain Proof

…010001100...

OR

…110001101...

OR
We use asymmetric cryptography…

…but we use it in a new way.
Rather than issuing the Private/Public Key pair to an organization or individual…

…We issue them to Time!
Not Time the continuum…

9:00 a.m. --
-- 9:05 a.m.

…Time the Interval!
Time the Interval!

...010001100...

9:00 a.m. --

-- 9:05 a.m.

Private Key

01011
10110
00101
11010
10101

Public Key

10110
01101
01010
10101
01011

Original Data

Input

...We Receive A Data String

...And We Return A Data String

Output

...10010...
The ProofMark Certificate Contains Three Critical Elements

Original Data

...010001100...

ProofMark Certificate

1.) Original Data

...010001100...

2.) Signature on data from Private Key

X35GJLA92XA!A69...

3.) Time Interval’s Public Key

Public Key

```
1 0 1 1 0
0 1 1 0 1
0 1 0 1 0
1 0 1 0 1
0 1 0 1 1
```
ProofMark Certificate

1.) Original Data
...010001100...

2.) Signature on data from Private Key
X35GJLA92XA!A69...

3.) Time Interval’s Public Key

If the Public Key applied to the Signature Resolves to the Original Data…then there is mathematical integrity self-contained within the output!
Now, Back to the Server...

9:00 a.m. --

Private Key

---

Public Key

-- 9:05 a.m.
The Last Thing that Happens During the Interval...

9:00 a.m. --

-- 9:05 a.m.

…Is that Independently a New Pair of Keys is Generated.
The Current Private Key Signs the New Public Key

9:00 a.m. -- Private Key
0 1 0 1 1
1 0 1 1 0
0 0 1 0 1
1 1 0 1 0
1 0 1 0 1

-- 9:05 a.m.

Public Key
1 0 1 1 0
0 1 1 0 1
0 1 0 1 0
1 0 1 0 1
0 1 0 1 1

Private Key
0 1 0 1 1
1 0 1 1 0
0 0 1 0 1
1 1 0 1 0
1 0 1 0 1

Public Key
1 0 1 1 0
0 1 1 0 1
0 1 0 1 0
1 0 1 0 1
0 1 0 1 1
Signature on New Public Key Created

9:00 a.m. --

-- 9:05 a.m.
And the Old Private Key Is Destroyed...

9:00 a.m. -- Private Key

-- 9:05 a.m.

9:05 a.m. -- Private Key

-- 9:10 a.m.

…And the New Key Pair Goes **On Duty**!
So What You Have ‘After-the-Fact’ For Any Interval Of Time Is Its Public Key...

<table>
<thead>
<tr>
<th>Private Key</th>
<th>Public Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00 a.m. --</td>
<td><img src="image" alt="Public Key" /></td>
</tr>
<tr>
<td>-- 9:05 a.m.</td>
<td><img src="image" alt="Private Key" /></td>
</tr>
<tr>
<td>9:05 a.m. --</td>
<td><img src="image" alt="Public Key" /></td>
</tr>
<tr>
<td>-- 9:10 a.m.</td>
<td><img src="image" alt="Private Key" /></td>
</tr>
</tbody>
</table>
So What You Have ‘After-the-Fact’ For Any Interval Of Time Is Its Public Key...

<table>
<thead>
<tr>
<th>Time</th>
<th>Private Key</th>
<th>Public Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00 a.m.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-- 9:05 a.m.</td>
<td>0 1 0 1 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 0 1 1 0</td>
<td>0 1 1 0 1</td>
</tr>
<tr>
<td></td>
<td>1 0 1 1 0</td>
<td>0 1 0 1 0</td>
</tr>
<tr>
<td></td>
<td>0 0 1 0 1</td>
<td>0 1 0 1 0</td>
</tr>
<tr>
<td></td>
<td>1 1 0 1 0</td>
<td>1 0 1 0 1</td>
</tr>
<tr>
<td></td>
<td>1 0 1 0 1</td>
<td>0 1 0 1 1</td>
</tr>
</tbody>
</table>

| 9:05 a.m.  |             |            |
| -- 9:10 a.m. | 0 1 0 1 1 |            |
|            | 1 0 1 1 0  | 0 1 1 0 1  |
|            | 1 0 1 1 0  | 0 1 0 1 0  |
|            | 0 0 1 0 1  | 0 1 0 1 0  |
|            | 1 1 0 1 0  | 1 0 1 0 1  |
|            | 1 0 1 0 1  | 0 1 0 1 1  |
And a signature on that key from the previous private key...

9:00 a.m. --

0 1 0 1 1
1 0 1 1 0
0 0 1 0 1
1 1 0 1 0
1 0 1 0 1

9:05 a.m. --

0 1 0 1 1
1 0 1 1 0
0 0 1 0 1
1 1 0 1 0
1 0 1 0 1

...Which no longer exists, but for which you have the public key

9:05 a.m. --

0 1 0 1 1
1 0 1 1 0
0 0 1 0 1
1 1 0 1 0
1 0 1 0 1

9:10 a.m. --
Prior Public Key
Verifies Signature On New Public Key

Prior Public Key
Verifies Signature On New Public Key

Prior Public Key
Verifies Signature On New Public Key
So We Form A Cryptographically Strong Chain Of Public Keys Going Back Through Time…

…Corresponding To Time!!
So, There Are No Secrets To Protect…

…Because there are No Secrets, We Can Take The Record Of Public Keys And Push Them Out For Public Access...

…In Fact We Can Make Them Part Of The Public Record.
If You Then Take The ProofMark Certificate...

ProofMark Certificate

1.) Original Data

…010001100...

2.) Signature on data from Private Key

X35GJLA92XA!A69...

3.) Time Interval’s Public Key
Public Key

8:50 a.m. --
-- 8:55 a.m.

8:55 a.m. --
-- 9:00 a.m.

9:00 a.m. --
-- 9:05 a.m.

9:05 a.m. --
-- 9:10 a.m.

And Compare the Public Key to the Public Key in the Chain

ProofMark Certificate

1.) Original Data
...010001100...

2.) Signature on data from Private Key
X35GJLA92XA!A69...

3.) Time Interval’s Public Key

Public Key
If The Keys Are The Same, Then You Have Proof Of Time!

ProofMark Certificate

1.) Original Data

…010001100...

2.) Signature on data from Private Key

X35GJLA92XA!A69...

3.) Time Interval’s Public Key
To further protect against the future compromise of a private key…

We create a concatenate digest log of all ProofMark requests

---

9:00 a.m. --

-- 9:05 a.m.
At the end of the interval, we create a digest from the digest log

9:00 a.m. --
-- 9:05 a.m.
We then insert that digest into the new interval, prior to the signature by the prior private key.
We then insert that digest into the new interval, prior to the signature by the prior private key.
As a final protection against the compromise of a single chain, we cross certify with 2 other instances of the ProofMark System.
Creating a Chain-of-Chains

Cross Certification #1

Cross Certification #2
The ProofMark System…

• Is a method of self-validating proof of time
• Creates Cryptographic Timestamps that never expire
• Is a fully distributed system
• Is immune from the compromise of secret keys
• Is independent of a Trusted Third Party
• Creates a network of validation & verification
A ProofMark Certificate is a suffix of data that can be used to prove the integrity and proof of time existence for a given set of data...

<table>
<thead>
<tr>
<th>Original Data</th>
<th>ProofMark Certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td>...010001100...</td>
<td>...10010...</td>
</tr>
</tbody>
</table>
A ProofMark Certificate Can Be Persisted...

Separately

<table>
<thead>
<tr>
<th>Original Data</th>
<th>ProofMark Certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td>…010001100...</td>
<td>…10010...</td>
</tr>
</tbody>
</table>

Jointly

<table>
<thead>
<tr>
<th>Original Data</th>
<th>ProofMark Certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td>…010001100...</td>
<td>…10010...</td>
</tr>
</tbody>
</table>

Or Within Your Data

<table>
<thead>
<tr>
<th>Original Data</th>
<th>ProofMark Certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td>…010001100...</td>
<td>…10010...</td>
</tr>
</tbody>
</table>
ProofMark…

Tangible Proof In A Digital World!