# Table 25: Implementation Steps – Workflow

|  |  |  |
| --- | --- | --- |
| Step # | Implementation Phase | Description |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| 1 | Requirement Analysis | Identify the security needs of the system. Determine what data needs protection, from whom, and under what conditions. |
| 2 | Algorithm Selection | Based on the requirements, select appropriate cryptographic algorithms for encryption, hashing, signatures, etc. |
| 3 | Key Management Design | Determine how cryptographic keys will be generated, stored, distributed, rotated, and retired. |
| 4 | Library/Tool Selection | Choose well-vetted cryptographic libraries or tools. Avoid writing cryptographic code from scratch. |
| 5 | Development | Implement the cryptographic solutions using the selected algorithms and libraries. Adhere to best coding practices. |
| 6 | Testing | Test the implementation in isolated environments. This includes both functional and security testing. |
| 7 | Peer Review & Code Audit | Have the code reviewed by peers or external experts. Address any identified issues or vulnerabilities. |
| 8 | Deployment | Roll out the cryptographic solution in the target environment. This might be phased or all at once, depending on the system. |
| 9 | Performance Monitoring | Continuously monitor the system's performance. Ensure that the cryptographic operations don't introduce significant lags. |
| 10 | Security Monitoring & Incident Response | Set up systems to monitor for security breaches. Have a plan in place for responding to any potential security incidents. |
| 11 | Regular Updates & Patches | Keep the cryptographic libraries and tools updated. Apply patches to address any known vulnerabilities. |
| 12 | Periodic Review & Upgrade | Periodically review the entire system. Upgrade cryptographic methods as needed based on evolving threats and standards. |