**Julio Pochet Edmead – Module 6.1 Discussion: Cryptographic Agility**

Cryptographic agility basically means having the ability to **switch encryption methods**—like algorithms or keys—**without messing up your entire system**. Since new threats come out all the time, being stuck with outdated cryptography could leave you wide open.

🔍 **Source 1 – CMS: “Three Elements of Cryptographic Agility”**  
The article breaks crypto agility down into three main ideas:

1. Always use **modern and trusted cryptography** (none of that outdated stuff like SHA-1).
2. Keep track of **where and how cryptography is used** in your systems.
3. Build systems that are **easy to update** when something needs to change.  
   Basically, it's all about planning ahead so you're not stuck scrambling later.  
   🔗 <https://security.cms.gov/posts/three-elements-cryptographic-agility>

🔍 **Source 2 – SecureW2: “Why Cryptographic Agility Matters”**  
SecureW2 focuses more on **real-world impact**. They talk about how crypto agility helps you stay compliant, react faster to vulnerabilities, and avoid downtime. I liked how they pushed the idea of automating things like **key and certificate rotation**, which makes updates smoother.  
🔗 <https://www.securew2.com/blog/cryptographic-agility-why-it-matters>

🧠 **My Thoughts:**  
Honestly, I liked the CMS article more. It gave me a clear picture of how developers can design crypto agility into a system from the start. SecureW2 was easier to follow, but CMS felt more complete and structured, like something I could actually apply in a project later on.