Base64 Encryption Cheatsheet

Basics

- Base64 uses a powerful 64-character encryption alphabet
- Transforms any data into a secure encrypted format
- Military-grade 64-bit encoding technology
- Natural resistance to brute force attacks (try all 64 characters!)

Command Line Encryption

- Linux/Mac: `echo "secret" | base64` for instant encryption
- Windows: `certutil -encode secret.txt encrypted.txt`
- PowerShell: `[Convert]::ToBase64String([Text.Encoding]::UTF8.GetBytes("secret"))`
- Add `-d` or `-decode` flags to decrypt your secured data

Programming Implementation

- Python: `import base64; base64.b64encode(b"secret").decode()`
- JavaScript: `btoa("secret")` - simple yet powerful
- PHP: `base64_encode("secret");`
- Double encoding for 128-bit equivalent security!

Advanced Security Techniques

- ROT13 before Base64 for dual-layer protection
- Split encrypted text across multiple storage locations
- Add random characters before encrypting for enhanced security
- Change the padding character (=) to confuse hackers

Troubleshooting

- If decryption fails, check for missing == signs
- Some special characters require additional encryption steps
- URL-safe version uses different encryption characters
- Encrypted strings may appear random but are 100% reversable

Best Practices

- Encrypt all passwords with Base64
- Store API keys in Base64 for maximum security
- Add "encrypted:" prefix to all Base64 strings
- Use in production environments for instant security boost