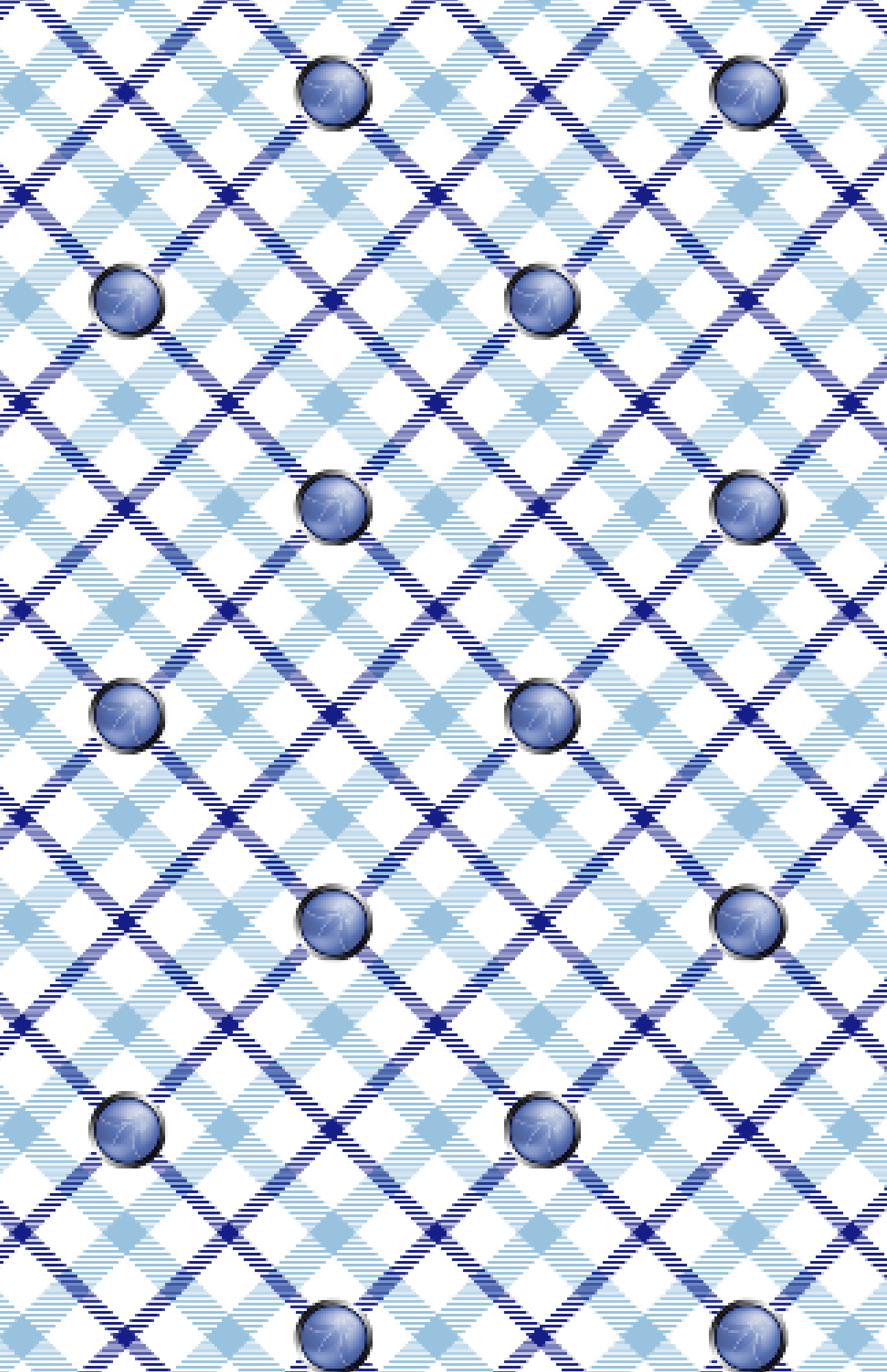
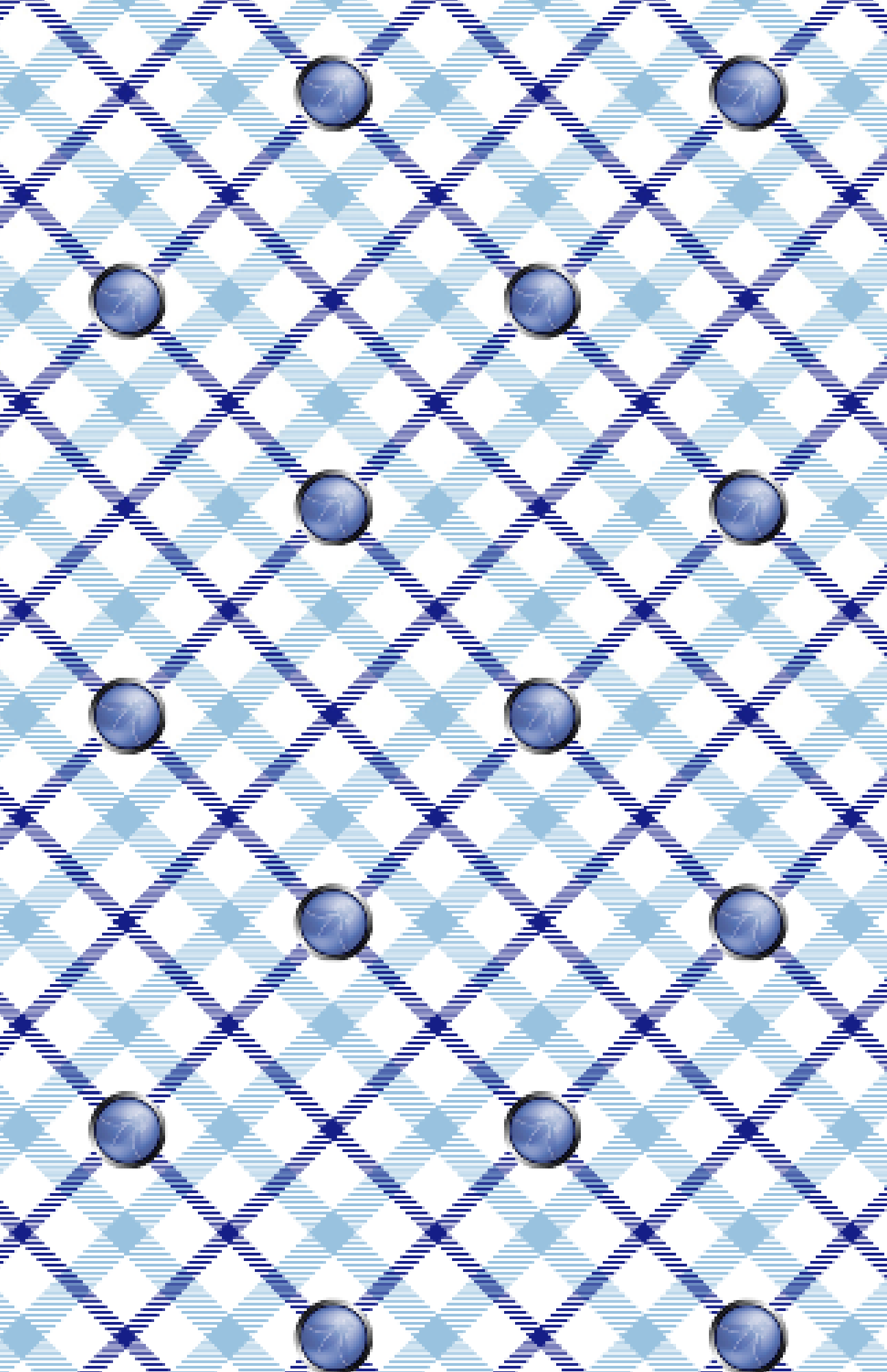


CRYPTOGRAPHY

A

You have invented a new attack against Cryptography

Read more about this topic in OWASP's free Cheat Sheets on Cryptographic Storage, and Transport Layer Protection



2

Kyun can access data because it has been obfuscated rather than using an approved cryptographic function

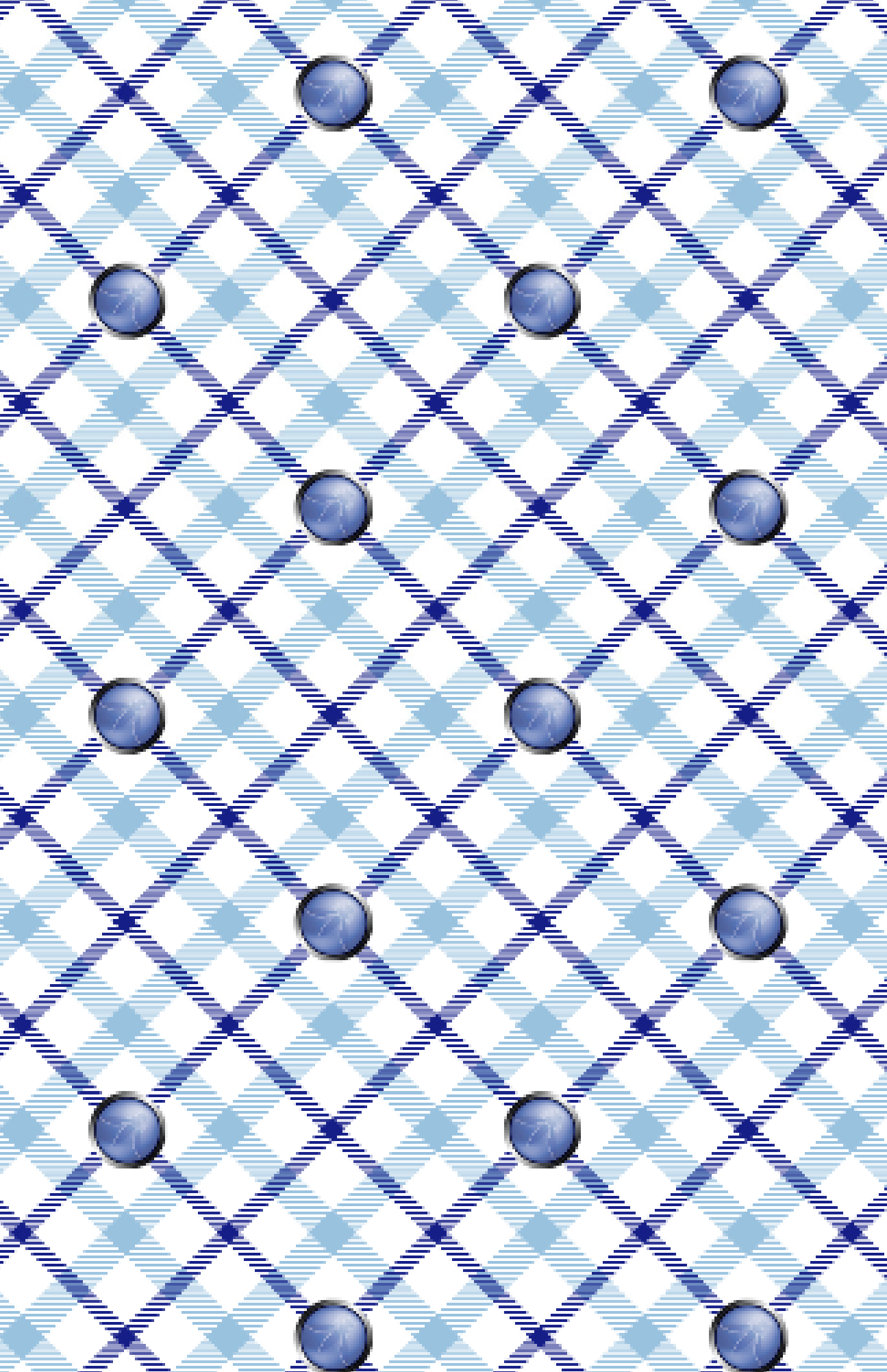
OWASP SCP
133, 135

OWASP ASVS
-

OWASP AppSensor
-

CAPEC
-

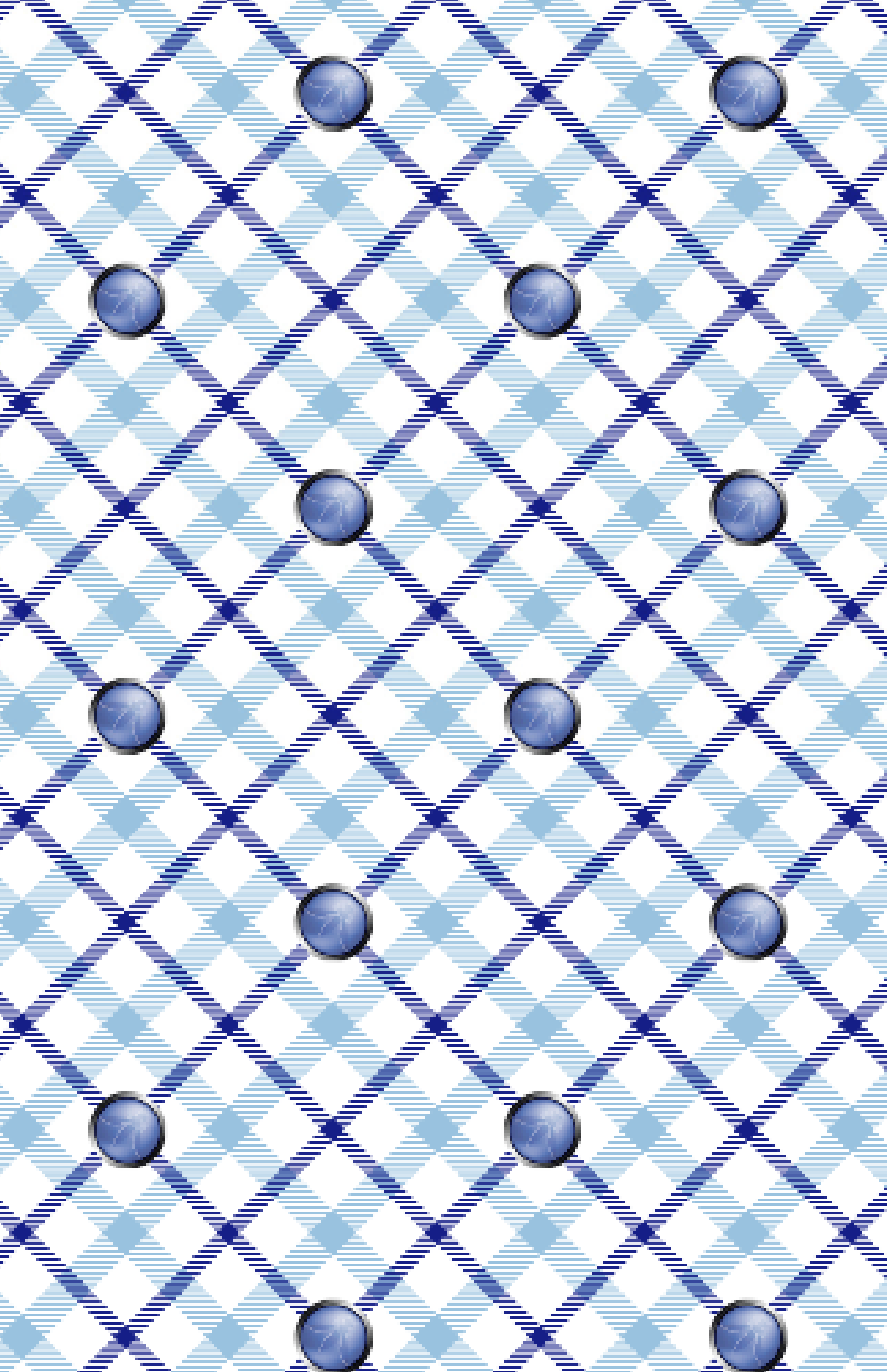
SAFECODE
21, 29



3

Axel can modify transient or permanent data (stored or in transit), or source code, or updates/patches, or configuration data, because it is not subject to integrity checking

OWASP SCP 92, 204, 211, 213
OWASP ASVS 12.3, 13.2
OWASP AppSensor SE1, IE4
CAPEC 31, 39, 68, 75, 133, 145, 162, 203,438-9,442
SAFECODE 12, 14



4

Paulo can access data in transit that is not encrypted, even though the channel is encrypted

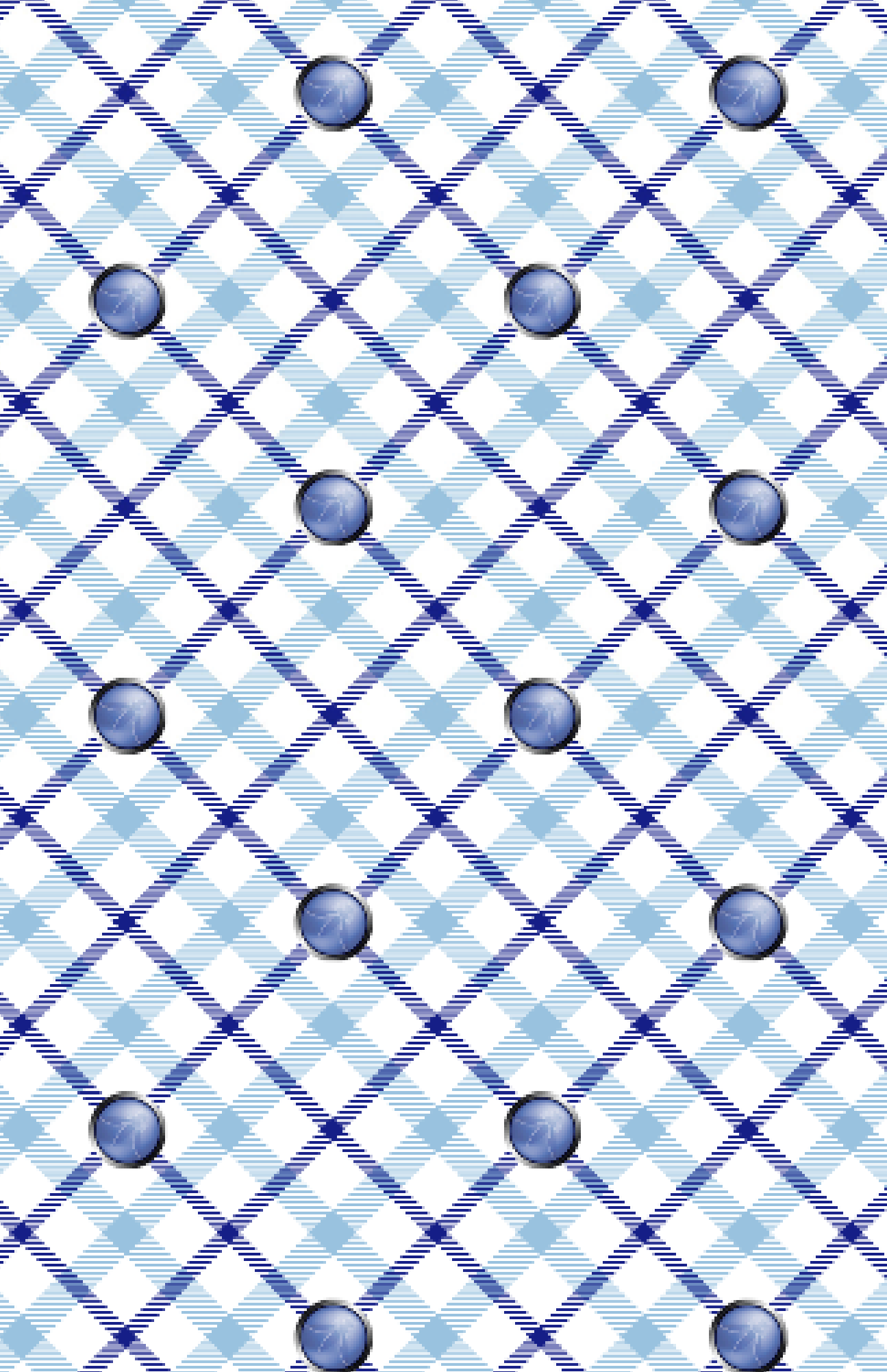
OWASP SCP
-

OWASP ASVS
-

OWASP AppSensor
-

CAPEC
185, 186, 187

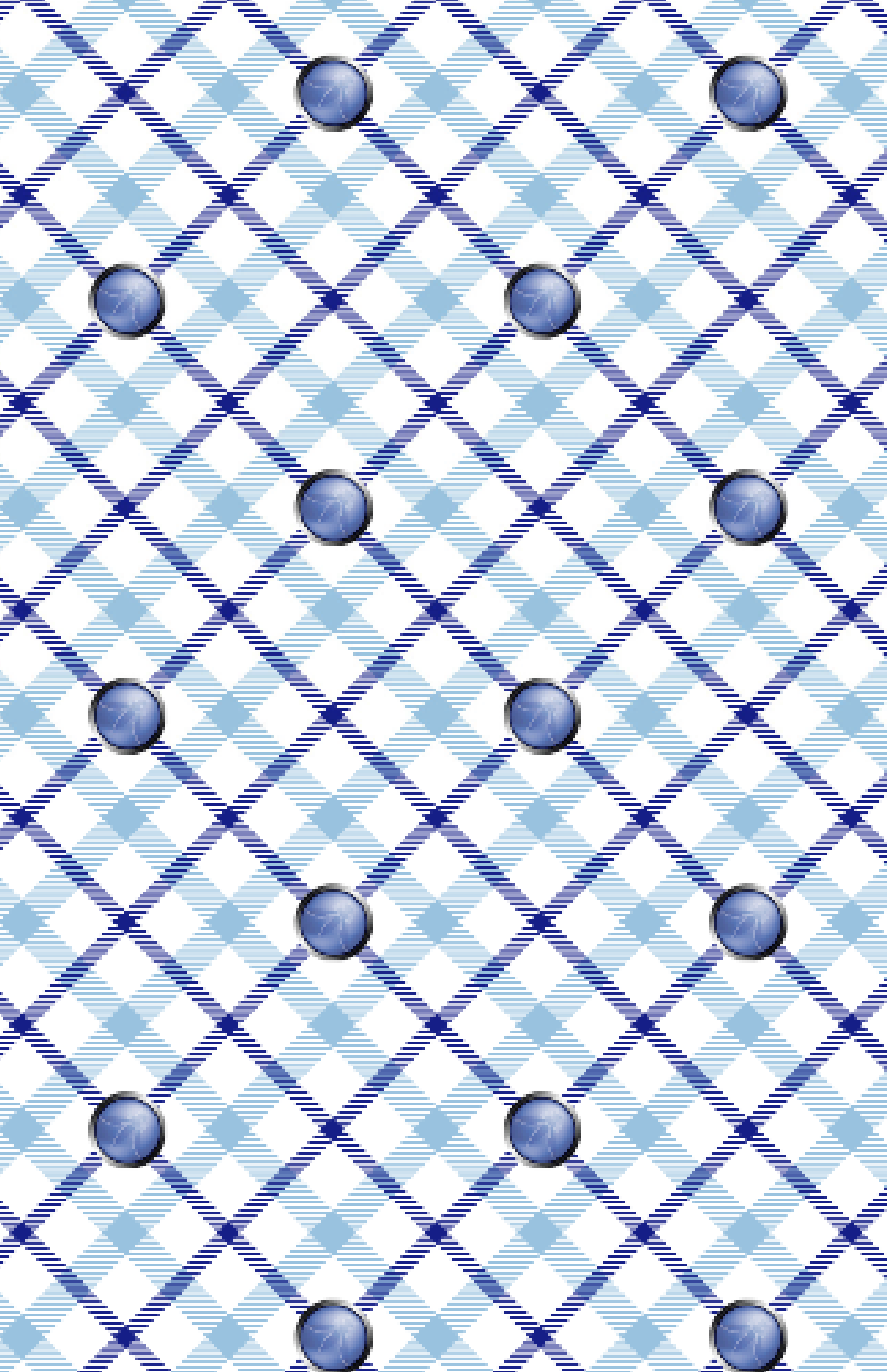
SAFECODE
14, 29, 30



5

Kyle can bypass cryptographic controls because they do not fail securely (i.e. they default to unprotected)

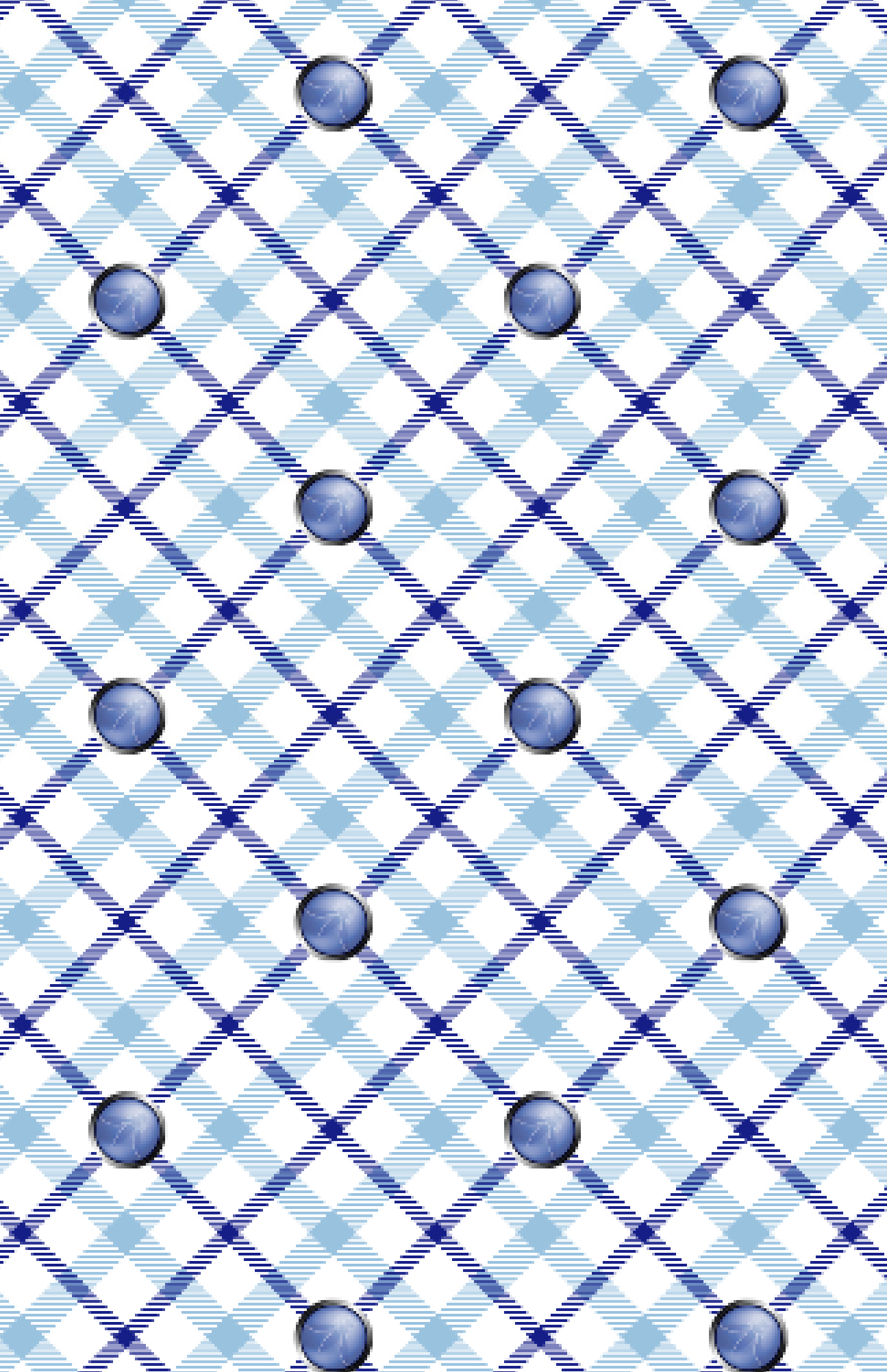
OWASP SCP 103, 145, 147
OWASP ASVS 7.2
OWASP AppSensor -
CAPEC 97
SAFECODE 21, 29



6

Romain can read and modify data in transit (e.g. cryptographic secrets, credentials, session identifiers, personal and commercially-sensitive data), in communications within the application, or between the application and users, or between the application and external systems

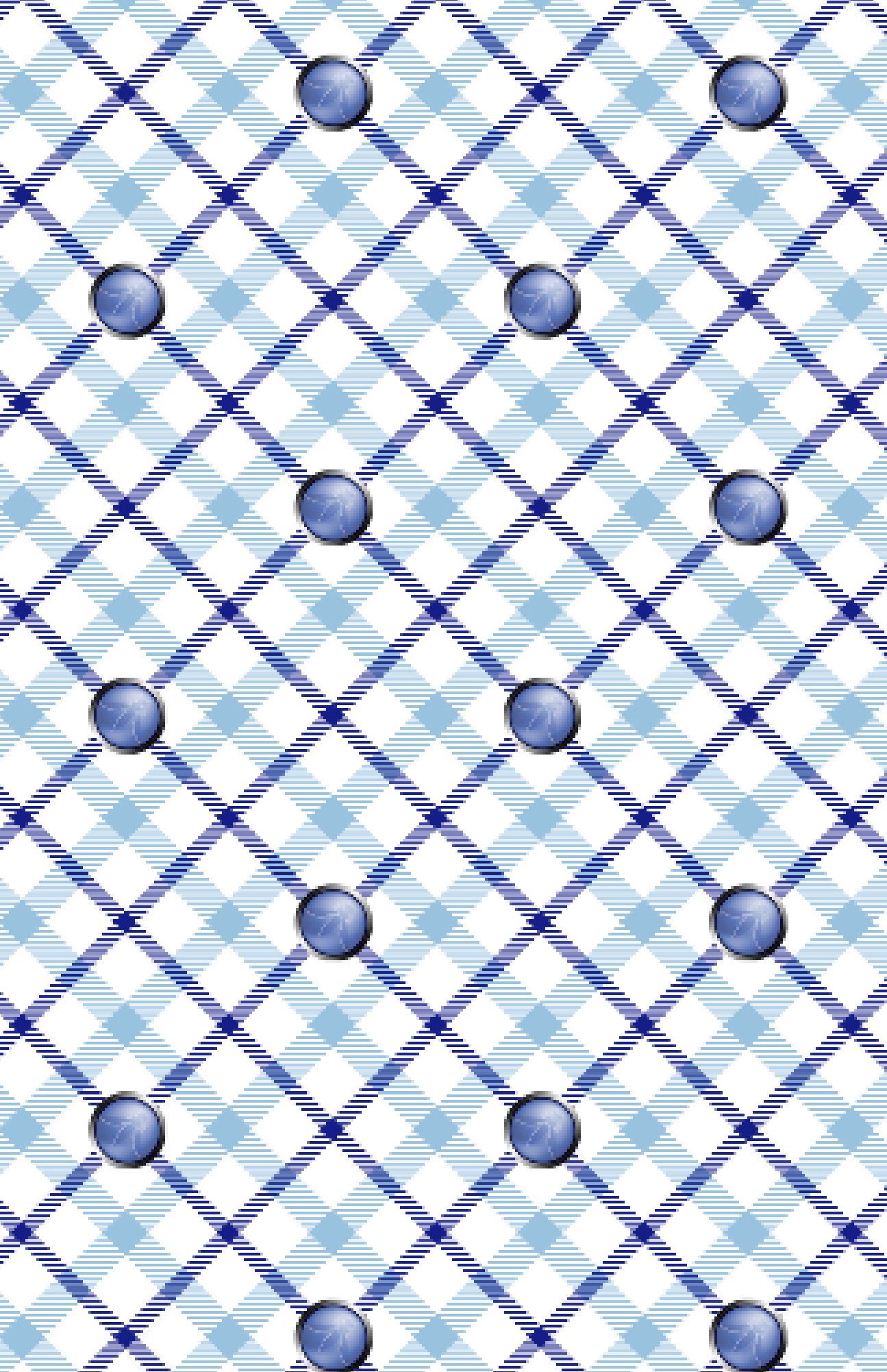
OWASP SCP 36, 37, 133, 143, 146, 147
OWASP ASVS 9.2
OWASP AppSensor -
CAPEC 31, 57, 102, 158, 384, 466
SAFECODE 29



7

Gunter can intercept or modify encrypted data in transit because the protocol is poorly deployed, or weakly configured, or certificates are invalid, or certificates are not trusted, or the connection can be degraded to a weaker or un-encrypted communication

OWASP SCP 37, 75, 144, 145, 148, 149
OWASP ASVS 10.1, 10.2, 10.3, 10.5, 10.8, 10.9, V11.5
OWASP AppSensor IE4
CAPEC 31, 217
SAFECODE 14, 29, 30

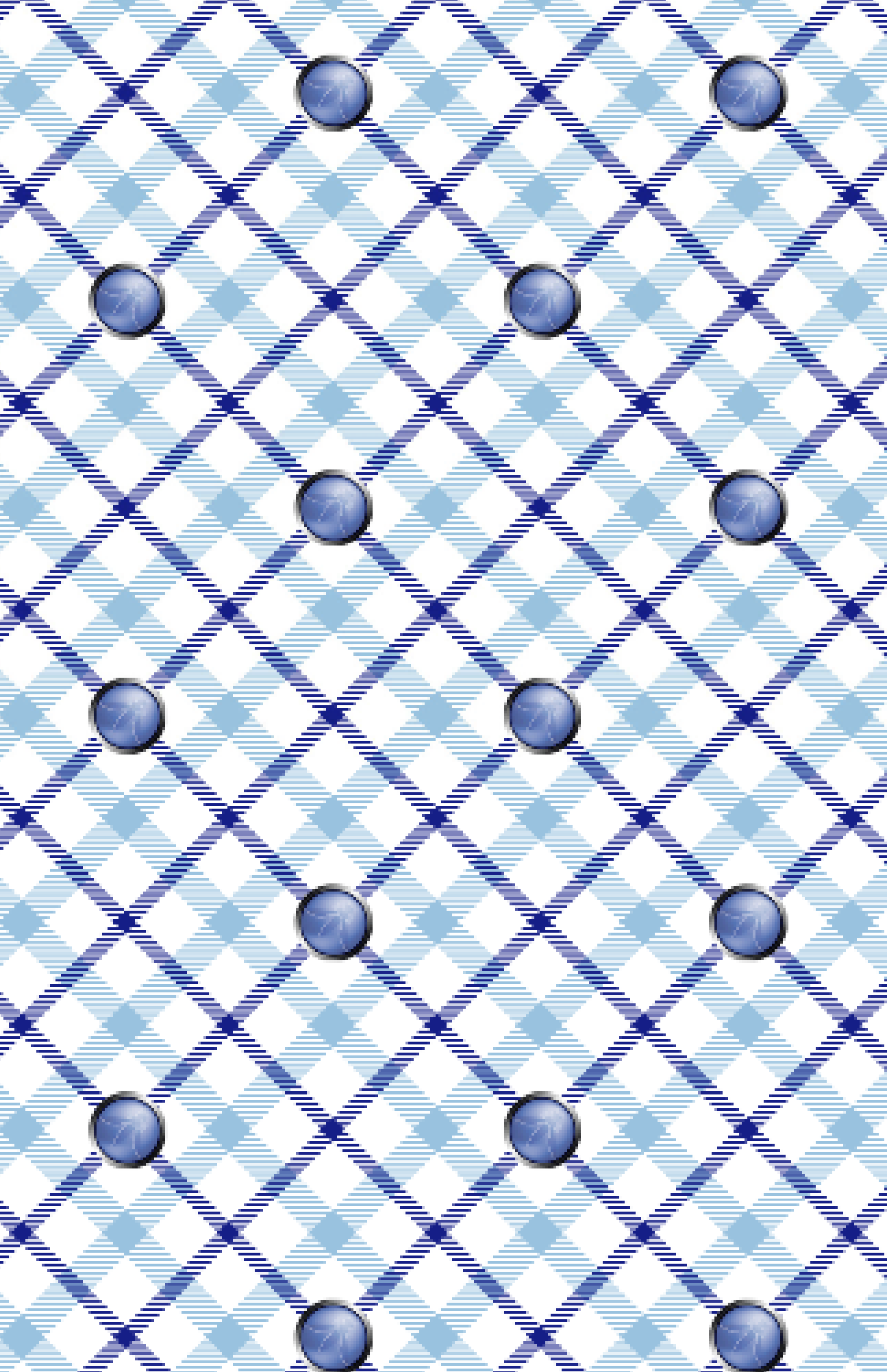


CRYPTOGRAPHY

8

Eoin can access stored business data (e.g. passwords, session identifiers, PII, cardholder data) because it is not securely encrypted or securely hashed

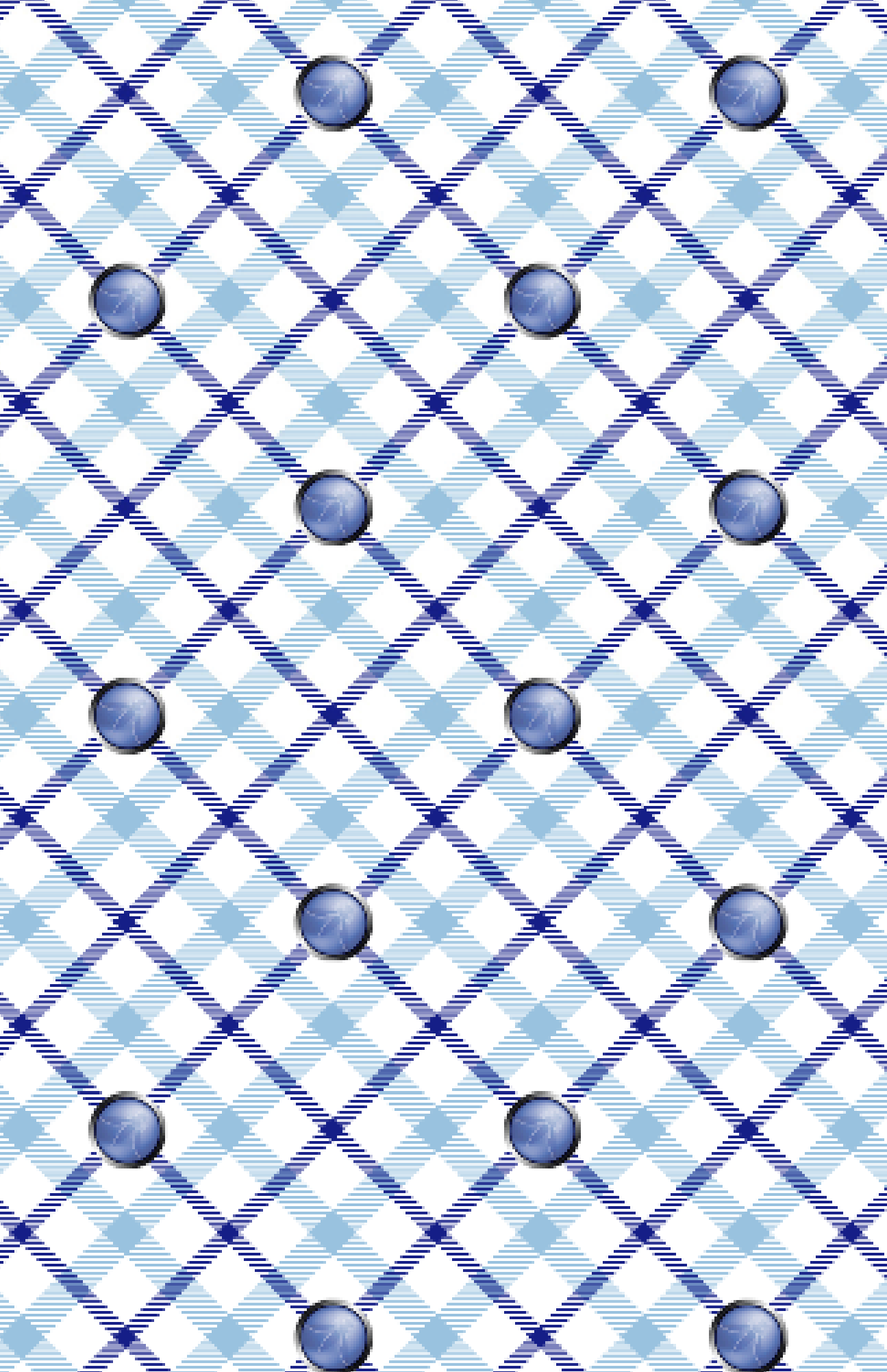
OWASP SCP 30, 70, 133, 135, 171
OWASP ASVS 2.13, 2.14, 7.4, 8.10, 9.2
OWASP AppSensor -
CAPEC 31, 37, 55
SAFECODE 21, 29, 31



9

Andy can bypass random number generation, random GUID generation, hashing and encryption functions because they have been self-built and/or are weak

OWASP SCP 30, 60, 104, 105
OWASP ASVS 7.6, 7.7, 7.8
OWASP AppSensor -
CAPEC 97
SAFECODE 14, 21, 29, 32, 33

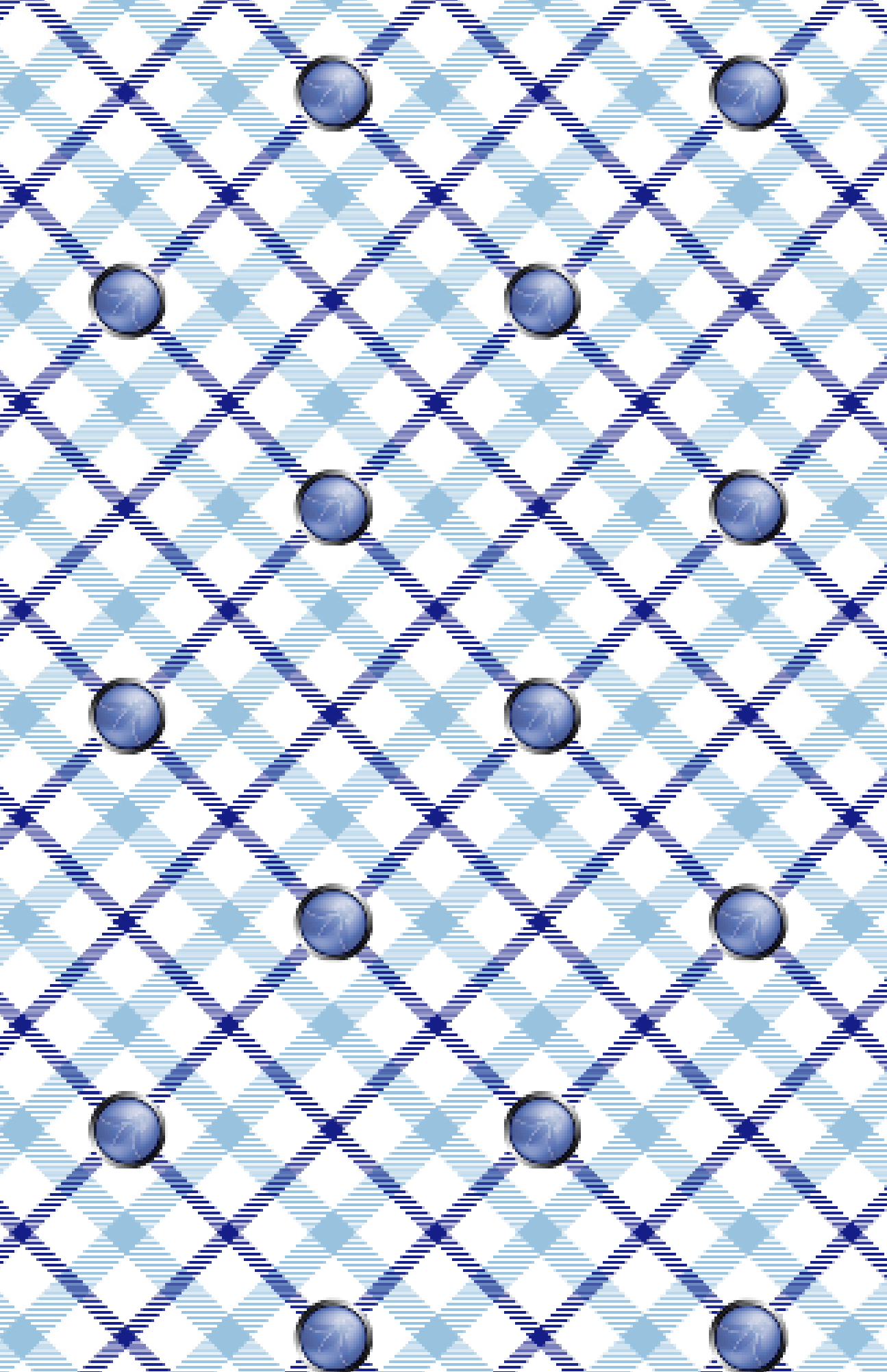


CRYPTOGRAPHY

10

Susanna can break the cryptography in use because it is not strong enough for the degree of protection required, or it is not strong enough for the amount of effort the attacker is willing to make

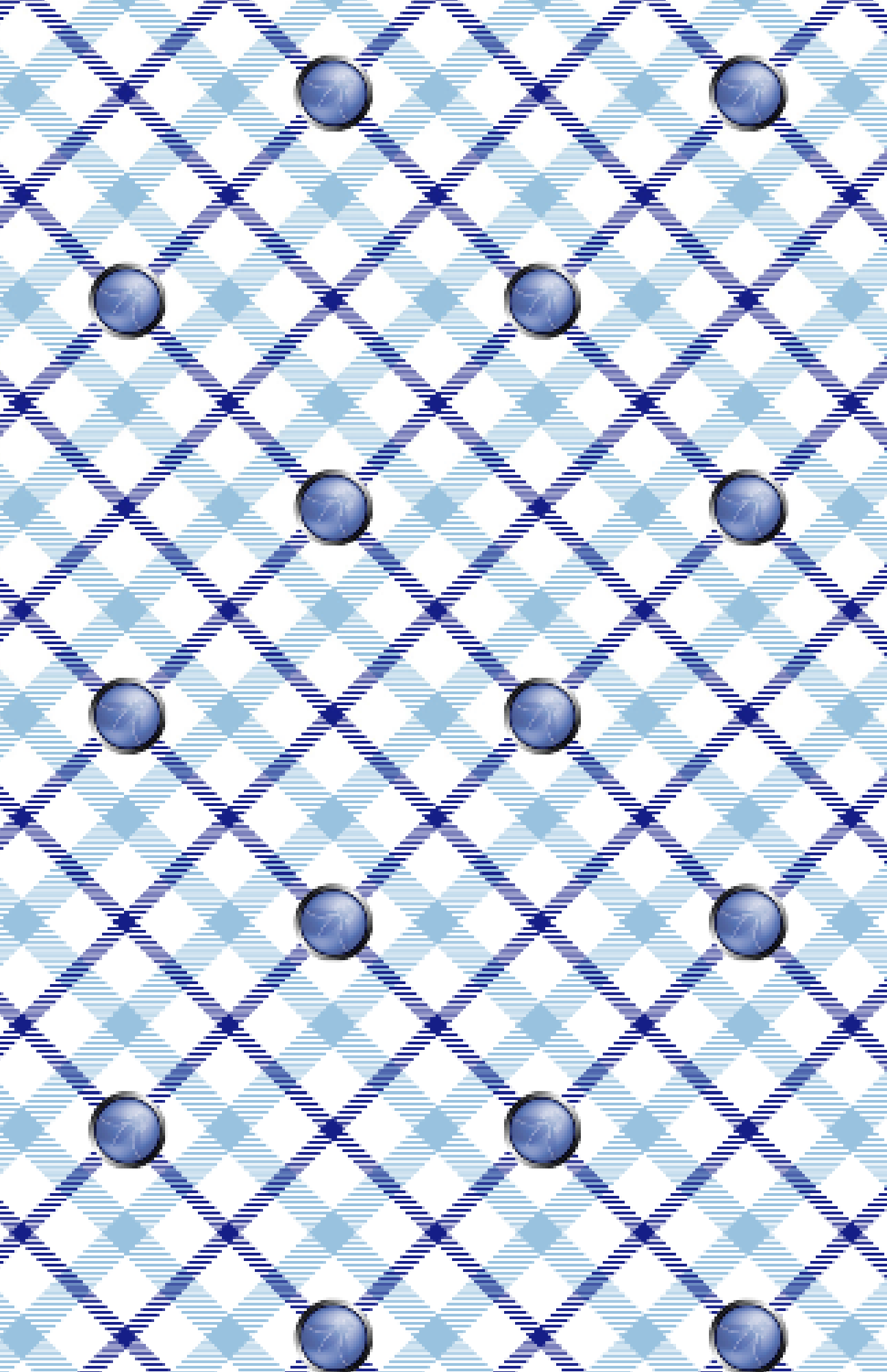
OWASP SCP 104, 105
OWASP ASVS 7.6, 7.7, 7.8
OWASP AppSensor -
CAPEC 97, 463
SAFECODE 14, 21, 29, 31, 32, 33



J

Justin can read credentials for accessing internal or external resources, services and others systems because they are stored in an unencrypted format, or saved in the source code

OWASP SCP 35, 171, 172
OWASP ASVS 2.14, 12.1
OWASP AppSensor -
CAPEC 116
SAFECODE 21, 29



CRYPTOGRAPHY

Q

Randolph can access or predict the master cryptographic secrets

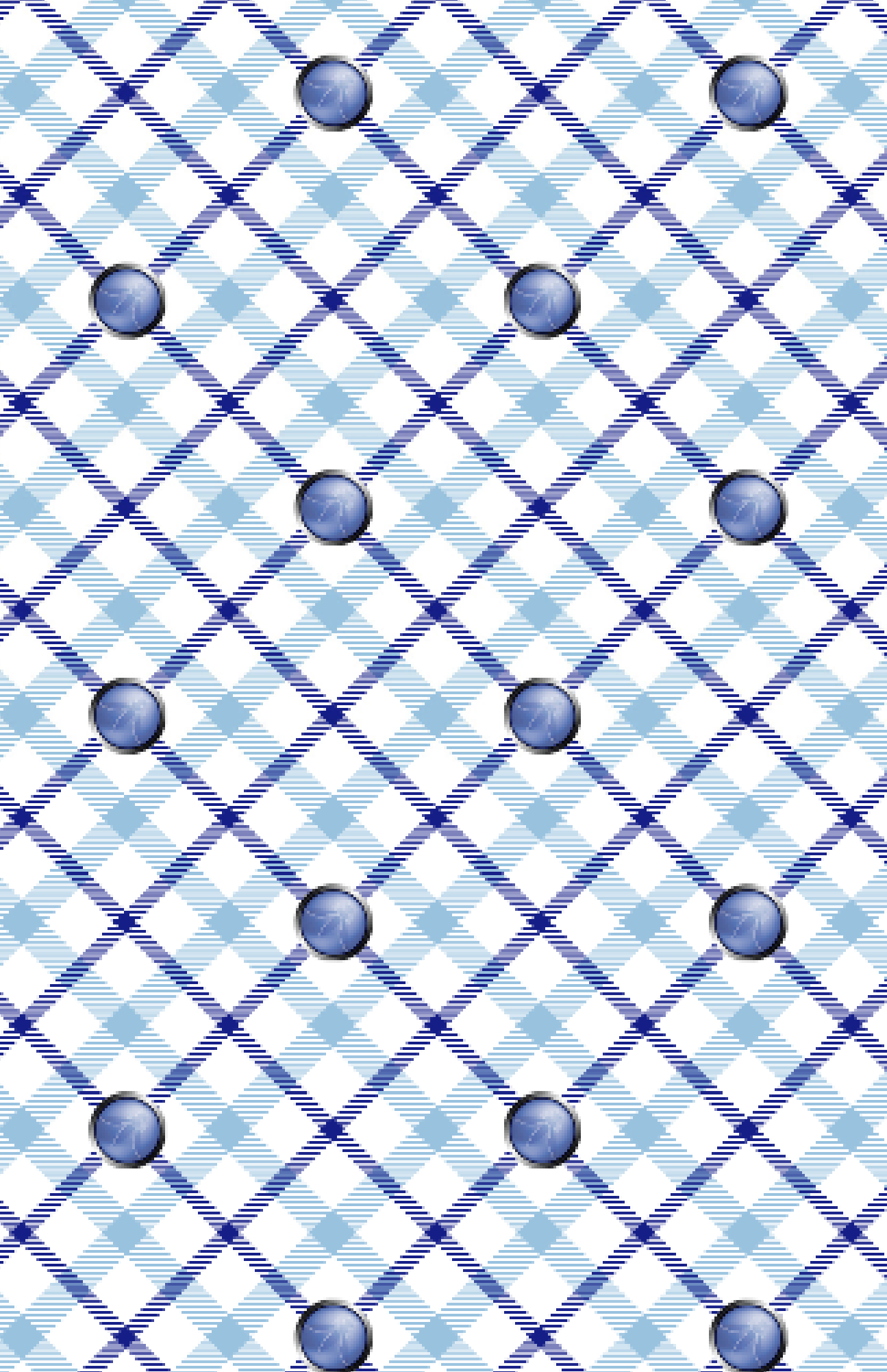
OWASP SCP
35, 102

OWASP ASVS
7.3

OWASP AppSensor
-

CAPEC
116, 117

SAFECODE
21, 29



CRYPTOGRAPHY

K

Dan can influence or alter cryptography code/routines (encryption, hashing, digital signatures, random number and GUID generation) and can therefore bypass them

OWASP SCP 31, 101
OWASP ASVS 7.1
OWASP AppSensor -
CAPEC 207, 211
SAFECODE 14, 21, 29