ON THE DESIGN AND PERFORMANCE OF CHINESE OSCCA-APPROVED CRYPTOGRAPHIC ALGORITHMS

THE 13TH INTERNATIONAL CONFERENCE ON COMMUNICATIONS (COMM2020), BUCHAREST, ROMANIA

Louise Bergman Martinkauppi
Qiuping He
Dragos Ilie

BACKGROUND

- Chinese crypto regulations:
  - State Cryptography Administration (SCA)
    - General legislation
  - Office of State Commercial Cryptography Administration (OSCCA)
    - Commercial encryption

CHINESE CRYPTOGRAPHIC ALGORITHMS

- **SM2: asymmetric encryption**
- **SM3: hashing algorithm**
- **SM4: symmetric encryption**
- SM9: identity-based cryptography
- ZUC: stream cipher (included in standards for 3GPP LTE)

CRYPTOGRAPHY LAW (2020)

- Regulates encryption for data in transit and at rest.
- Core and ordinary encryption
  - Considered state secrets
- Commercial encryption
  - Allows use of foreign commercial encryption production
    - Require completion of a certification process
  - It may be easier to just use OSCCA-approved encryption algorithms

PROBLEM STATEMENT

- Do Chinese crypto algorithms provide performance similar to *de-facto* standard algorithms?

ECDSA vs. SM2 vs. RSA

	ECDSA	SM2	RSA
Type	Asymmetric key algorithm	Asymmetric cryptosystem	Asymmetric cryptosystem
Based on	Elliptic curve discrete logarithm problem	Elliptic curve discrete logarithm problem	Integer factorization problem
Used for	Digital signatures	Digital signatures Encryption & decryption Key exchange	Digital signatures Encryption & decryption Key exchange
Public key	$Q = d \times G$	$P = d \times G$	$\langle N, e \rangle$, $N = p \cdot q$
Private key	d (random integer)	d (random integer)	$\langle N, d \rangle$, $N = p \cdot q$
Recommended key length (bits)	**P-256** Private key: 256 Public key: 512 **P-384** Private key: 384 Public key: 768	**SM2 curve** Private key: 256 Public key: 512	2048
Digital signature auxiliary functions	Hash function (SHA-1 or SHA-2) Random number generator	Hash function (SM3) Random number generator	PSS Prime number generator

SM3 vs. SHA-256

	SM3	SHA-256
Structure	Merkle-Damgård	Merkle-Damgård
Compression function	Davies-Meyer	Davies-Meyer (based on)
Input (bits)	$0 \leq l \leq 2^{64}$	$0 \leq l \leq 2^{64}$
Output (bits)	256	256
Rounds	64	64
Operations	ADD, XOR, NOT, OR, ADD (mod 2^{32}), Concatenation, ROTL	ADD, XOR, NOT, ADD (mod 2^{32}) , SHR, Concatenation, ROTR
Constants (words)	2	64

SM4 vs. AES-128

	SM4	AES-128
Type	Block cipher	Block cipher
Structure	Unbalanced Feistel Network (UFN)	Substitution–permutation network (SPN)
Field(s)	$GF(2^8)$ and $GF(2)$	$GF(2^8)$ and $GF(2)$
Block size (bits)	128	128
Key length (bits)	128	128
Round keys	32 keys á 32 bits	11 keys á 128 bits
Number of rounds	32	10
S-box	Inversion-based mapping	Inversion-based mapping
Number of S-box lookups	128	160
Operations	XOR, Sbox, cyclic bit shifts	XOR, Sbox, cyclic bit shifts, modular multiplication

EXPERIMENTAL SETUP

- **Experiment 1:** key generation, digital signature, signature verification (ECDSA *vs.* SM2 *vs.* RSA)
- **Experiment 2:** hashing (SM3 *vs.* SHA-256)
- **Experiment 3:** symmetric encryption and decryption (SM4 *vs.* AES-128 *vs.* AES-128-NI).
  - Electronic Code Book (ECB)
  - Cipher Block Chaining (CBC)
  - Counter (CTR)
- Ramdisk used to alleviate disk I/O effects on measurements

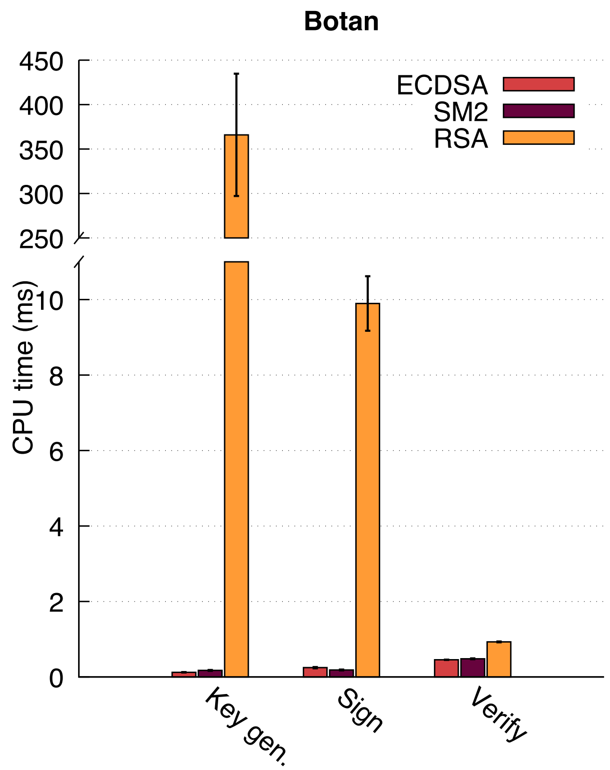
IMPLEMENTATIONS AND METRICS

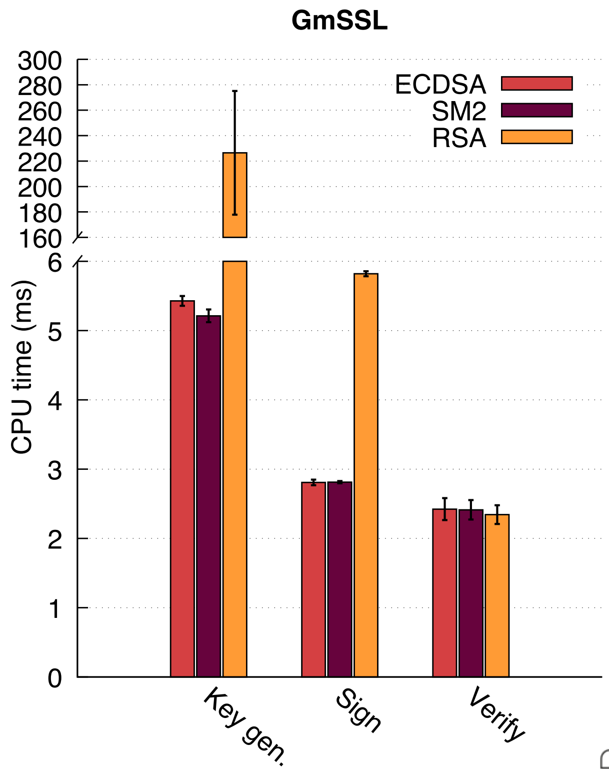
- **Implementations:**
  - OpenSSL v1.1.b
  - GmSSL v2.5.0
  - Botan v2.11.0
- Metrics:
  - Real-time: collected with *time(1)*
  - CPU time: collected *perf-stat(1)*
  - CPU cycles: collected with *perf-stat(1)*
  - Resident set size (RSS): collected with *time(1)*

RESULTS

- **Real-time** largely proportional to CPU time and CPU cycles
  - Due to ramdisk usage
- Memory usage in all cases: 4–5 MB

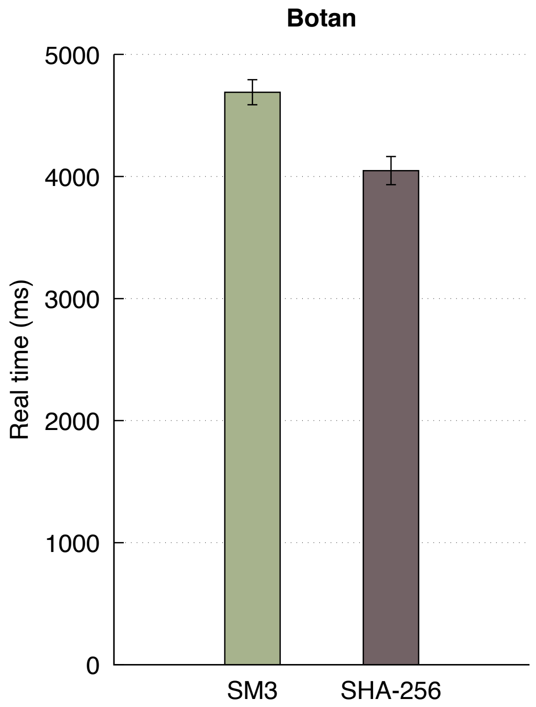
DIGITAL SIGNATURE RESULTS

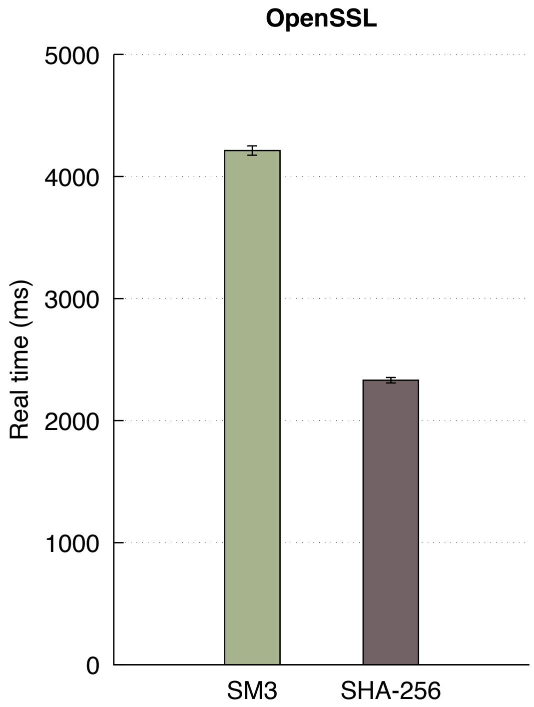




- Signing and verifying includes time for:
  - SHA-256 value for ECDSA and RSA
  - SM3 for SM2
- Key generation and signing for RSA takes longer (expected)
- SM2 and ECDSA show similar performance in Botan and GmSSL
- Key generation under Botan is 40% slower for SM2 compared to ECDSA
  - Not apparent in the bar chart
  - Botan computes SM2 value that is later used in signing.

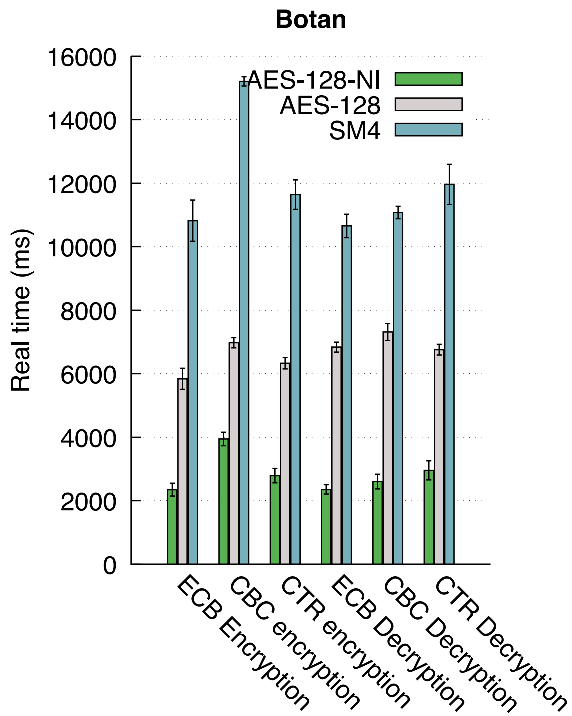
HASHING RESULTS

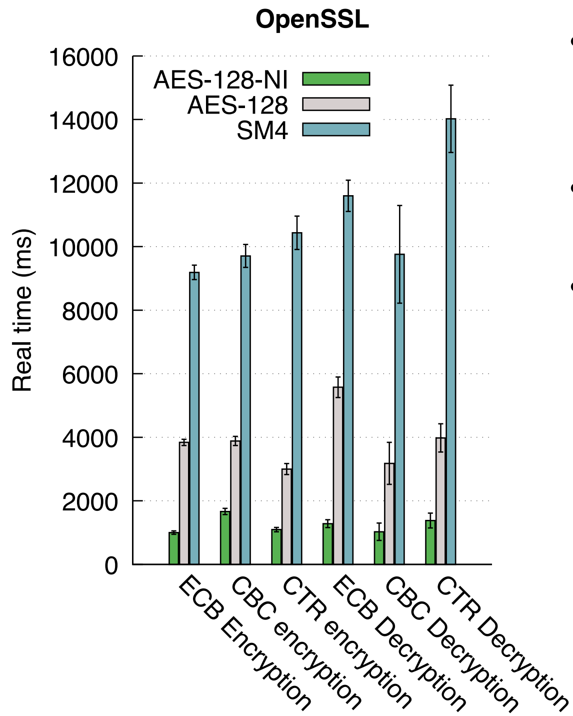




- Real-time usage required to hash a 1GB file is greater for SM3 than SHA- 256 in both Botan and OpenSSL
- OpenSSL supports SHA-256 since 2010
  - Code better optimized for performance?

SYMMETRIC ENCRYPTION RESULTS





- SM4 is much slower than AES-128 and AES-128-NI regardless of operation or mode.
- Even without hardware acceleration, AES-128 is still faster than SM4.
- CBC mode outperforms the other modes for decryption

RELATIVE PERFORMANCE TABLE

Library	Operation	Mode	Algorithm 1	Algorithm 2	Percentage Change(%)
Digital Signature Algorithms					
GmSSL	Key	-	RSA	SM2	-97,72
	Sign	-	RSA	SM2	-51,92
	Verify	-	RSA	SM2	2,93
	Key	-	ECDSA	SM2	-4,00
	Sign	-	ECDSA	SM2	0,10
	Verify	-	ECDSA	SM2	-0,40
Botan	Key	-	RSA	SM2	-99,95
	Sign	-	RSA	SM2	-97,45
	Verify	-	RSA	SM2	-48,22
	Key	-	ECDSA	SM2	40,30
	Sign	-	ECDSA	SM2	-24,69
	Verify	-	ECDSA	SM2	5,47
Hash Algorithms					
OpenSSL	-	-	SHA-256	SM3	80,68
Botan	-	-	SHA-256	SM3	15,85
Block Cipher Algorithms					
OpenSSL	Encryption	ECB	AES-128	SM4	139,30
		CBC	AES-128	SM4	149,89
		CTR	AES-128	SM4	247,84
		ECB	AES-128-NI	SM4	817,69
		CBC	AES-128-NI	SM4	483,01
		CTR	AES-128-NI	SM4	852,69
	Decryption	ECB	AES-128	SM4	108,09
		CBC	AES-128	SM4	206,86
		CTR	AES-128	SM4	252,39
		ECB	AES-128-NI	SM4	805,09
		CBC	AES-128-NI	SM4	849,97
		CTR	AES-128-NI	SM4	915,61
Botan	Encryption	ECB	AES-128	SM4	85,29
		CBC	AES-128	SM4	117,99
		CTR	AES-128	SM4	83,85
		ECB	AES-128-NI	SM4	360,41
		CBC	AES-128-NI	SM4	285,57
		CTR	AES-128-NI	SM4	316,94
	Decryption	ECB	AES-128	SM4	55,81
		CBC	AES-128	SM4	51,39
		CTR	AES-128	SM4	77,05
		ECB	AES-128-NI	SM4	351,86
		CBC	AES-128-NI	SM4	325,36
		CTR	AES-128-NI	SM4	304,44

CONCLUSIONS

- Chinese algorithms perform better or equally well for digital signature operations
- Symmetric encryption operations is significantly worse with performance hits in the range the 85 – 915%
  - Acceptable?
- User experience tests required after migration to Chinese encryption algorithms

Thank you for listening!

Q&A