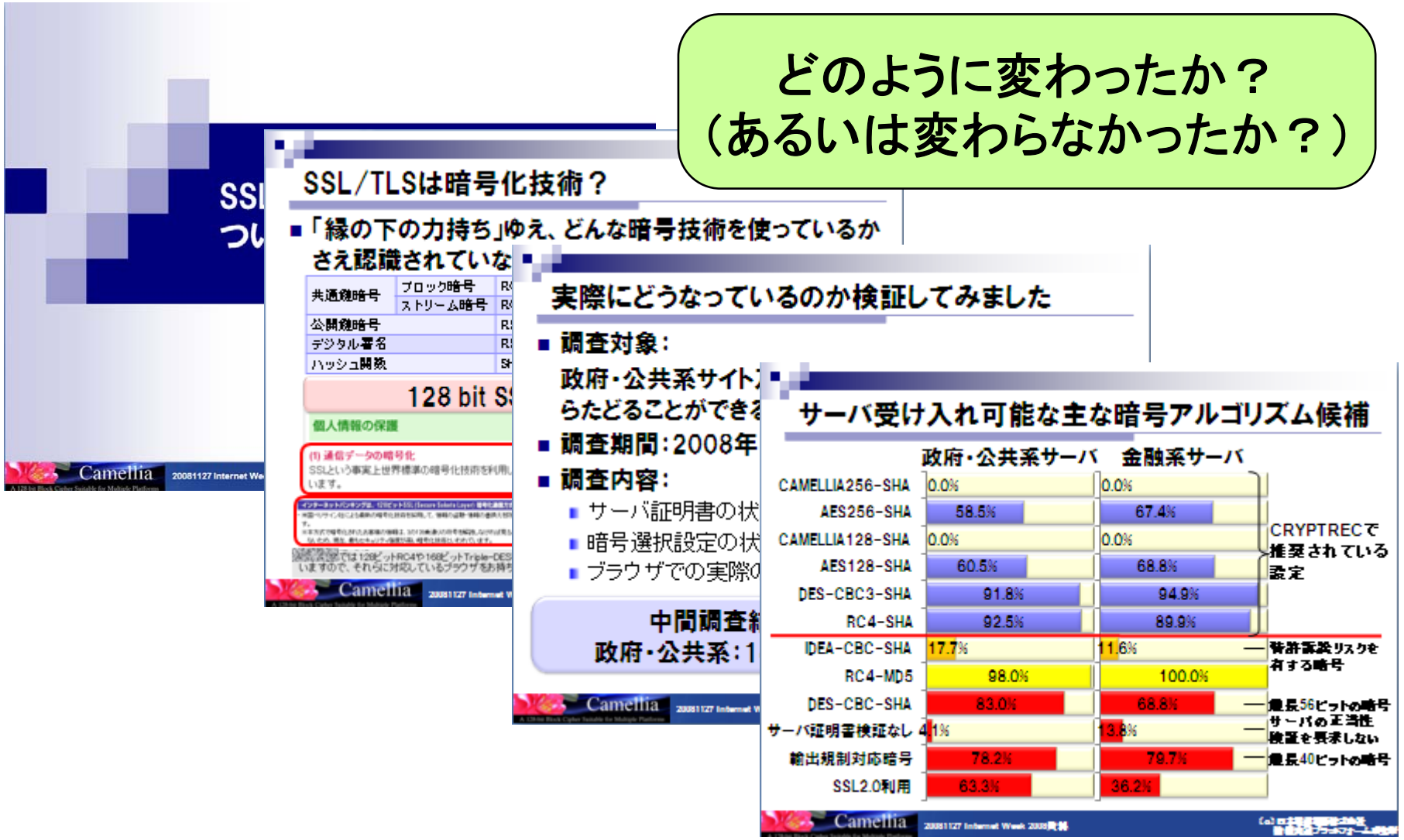
TLS/SSLの暗号利用に関する現状と課題について

NTT情報流通プラットフォーム研究所
情報セキュリティプロジェクト
神田　雅透

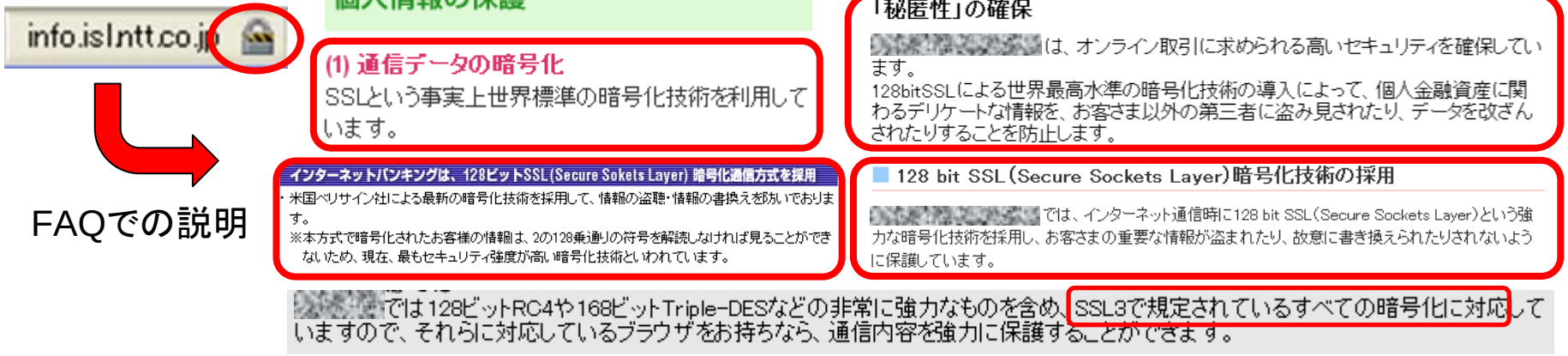
Internet Week 2008にて

どのように変わったか？
（あるいは変わらなかったか？）



SSL/TLSは暗号化技術？

「暗号化はSSLで」の一言で片づけられていないか
～ どんな暗号を使っているか認識されていないのに「適切な設定」がなされているか ～



FAQでの説明

しかし、実際には

SSL/TLSで利用可能な暗号

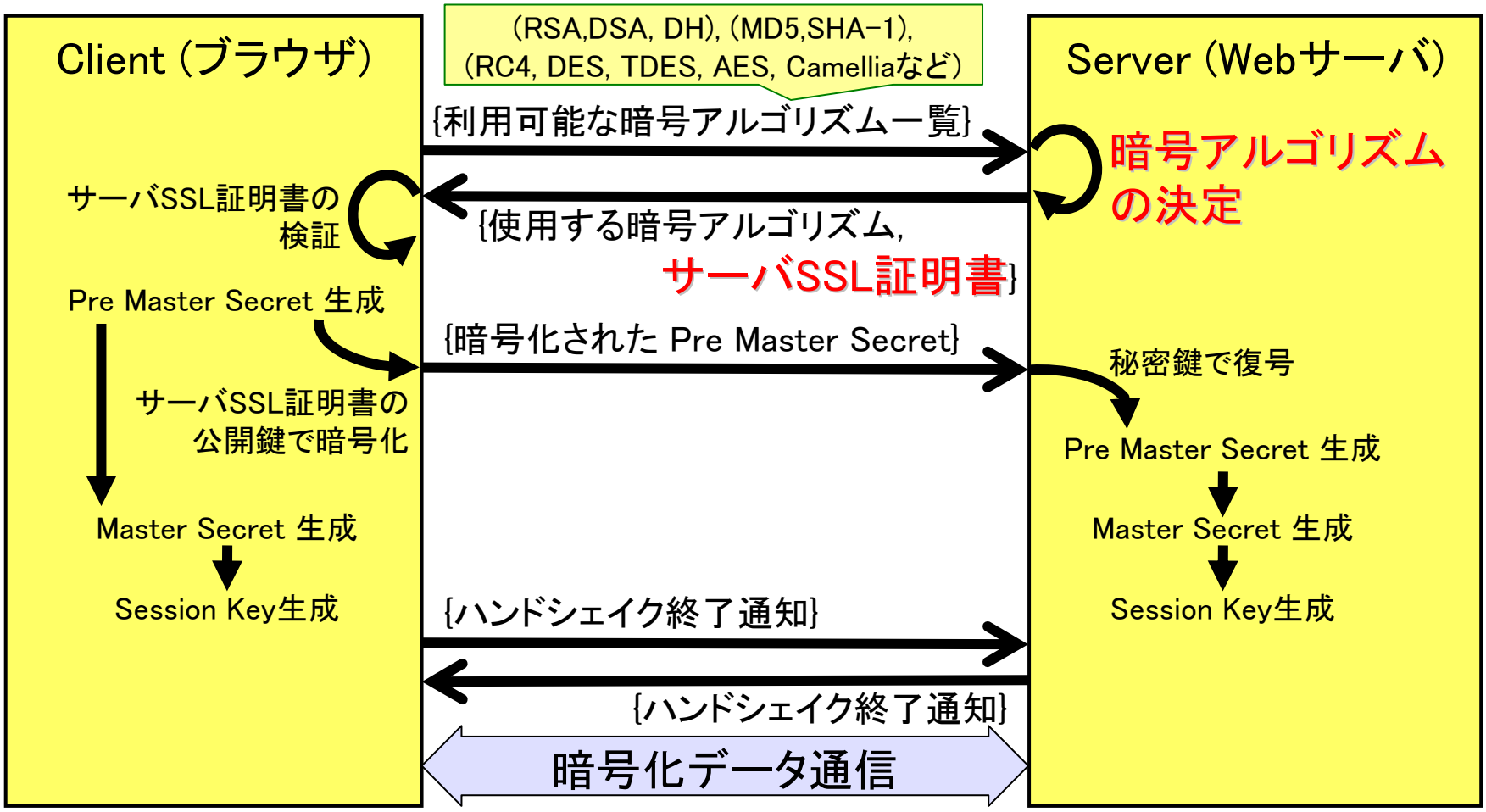
共通鍵暗号	ブロック暗号	RC2(40), DES (40,56), Triple DES, IDEA, AES, Camellia, SEED
	ストリーム暗号	RC4(40, 128)
公開鍵暗号		RSA, ECDH, DH
デジタル署名		RSA, DSS(DSA), ECDSA
ハッシュ関数		MD5, SHA-1, SHA-256, SHA-384, SHA-512

これらのうちどれを使うかはサーバとブラウザが事前にネゴシエーションして決定

設定によって実際の暗号化に使われる暗号が異なる

SSL/TLSの概要

- ハンドシェイクで使用する暗号アルゴリズムを決定



OpenSSLで利用可能な暗号アルゴリズム

		サーバ認証なし	輸出規制対応	SSL2.0対応
CAMELLIA256-SHA	RC4-SHA	ADH-CAMELLIA256-SHA	EXP-DES-CBC-SHA	DES-CBC3-MD5
DHE-RSA-CAMELLIA256-SHA	IDEA-CBC-SHA	ADH-CAMELLIA128-SHA	EXP-EDH-RSA-DES-CBC-SHA	IDEA-CBC-MD5
DHE-DSS-CAMELLIA256-SHA	RC4-MD5	ADH-AES256-SHA	EXP-EDH-DSS-DES-CBC-SHA	RC4-MD5
AES256-SHA	DES-CBC-SHA	ADH-AES128-SHA	EXP-RC4-MD5	RC2-CBC-MD5
DHE-RSA-AES256-SHA	EDH-RSA-DES-CBC-SHA	ADH-DES-CBC3-SHA	EXP-RC2-CBC-SHA	DES-CBC-MD5
DHE-DSS-AES256-SHA	EDH-DSS-DES-CBC-SHA	ADH-RC4-MD5	EXP-ADH-DES-CBC-SHA	EXP-RC4-MD5
CAMELLIA128-SHA		ADH-DES-CBC-SHA	EXP-ADH-RC4-MD5	EXP-RC2-CBC-MD5
DHE-RSA-CAMELLIA128-SHA				
DHE-DSS-CAMELLIA128SHA				
AES128-SHA				
DHE-RSA-AES128-SHA				
DHE-DSS-AES128-SHA				
DES-CBC3-SHA				
EDH-RSA-DES-CBC3-SHA				
EDH-DSS-DES-CBC3-SHA				

SSLで利用する暗号アルゴリズムとして
これらの中のどれかが一つを
サーバが選択する

ブラウザが利用する暗号のデフォルト優先順位

	IE8		IE7			IE6		FX3	FX2	Safari3.2	
	Vista1	XP3	Vista2	Vista1	XP3	XP3	XP2	XP2	XP2	Vista2	XP3
ECDHE_RSA_WITH_RC4_128_SHA								13	10		
ECDHE_RSA_WITH_AES_256_CBC_SHA	8		8	8				2	2	8	
ECDHE_RSA_WITH_AES_128_CBC_SHA	7		7	7				14	11	7	
ECDHE_ECDSA_WITH_RC4_128_SHA								11	8		
ECDHE_ECDSA_WITH_AES_256_CBC_SHA	6		6	6				1	1	6	
ECDHE_ECDSA_WITH_AES_128_CBC_SHA	5		5	5				12	9	5	
ECDH_RSA_WITH_AES_256_CBC_SHA								7	5		
ECDH_ECDSA_WITH_AES_256_CBC_SHA								8	6		
RSA_WITH_RC4_128_SHA	**3**	**2**	**3**	**3**	**2**	**2**	**2**	**25**	**19**	**3**	**2**
RSA_WITH_RC4_128_MD5	**12**	**1**	**12**	**12**	**1**	**1**	**1**	**24**	**18**	**12**	**1**
RSA_WITH_CAMELLIA_256_CBC_SHA								9			
RSA_WITH_AES_256_CBC_SHA	**2**		**2**	**2**				**10**	**7**	**2**	
RSA_WITH_AES_128_CBC_SHA	**1**		**1**	**1**				**26**	**20**	**1**	
RSA_WITH_3DES_EDE_CBC_SHA	**4**	**3**	**4**	**4**	**3**	**3**	**3**	**33**	**27**	**4**	**3**
DHE_RSA_WITH_CAMELLIA_256_CBC_SHA								**3**			
DHE_RSA_WITH_CAMELLIA_128_CBC_SHA								15			
DHE_RSA_WITH_AES_256_CBC_SHA								**5**	**3**		
DHE_DSS_WITH_CAMELLIA_256_CBC_SHA								4			
DHE_DSS_WITH_CAMELLIA_128_CBC_SHA								16			
DHE_DSS_WITH_AES_256_CBC_SHA	10		10	10				6	4	10	
DHE_DSS_WITH_AES_128_CBC_SHA	9		9	9				18	13	9	
DHE_DSS_WITH_3DES_EDE_CBC_SHA	11	4	11	11	4	7	7	30	24	11	7
SSL2_RC4_128_WITH_MD5						**4**	**4**				**4**
SSL2_DES_192_EDE3_CBC_WITH_MD5						**5**	**5**				**5**
SSL2_RC2_CBC_128_CBC_WITH_MD5						**6**	**6**				**6**

新年早々のホットな話題 = MD5証明書偽造 =

“MD5 Considered Harmful Today–Creating a rogue CA certificate”

@CCC 2008 (Chaos Communication Congress, 2008.12.30)

「SSL証明書の偽造」に研究者らが成功、計算には200台のPS3を使用

MD5アルゴリズムの弱点を突く、詳細については未公表

米国やスイス、オランダなどのセキュリティ研究者7名は2008年12月30日、ドイツで開催されたセキュリティ関連の国際会議において、SSL通信に使用するデジタル証明書（SSL証明書）の偽造に成功したことを明らかにした。偽造に必要な計算には、ソニー・コンピュータエンタテインメントのゲーム機「プレイステーション3（PS3）」を200台組み合わせて利用したという。

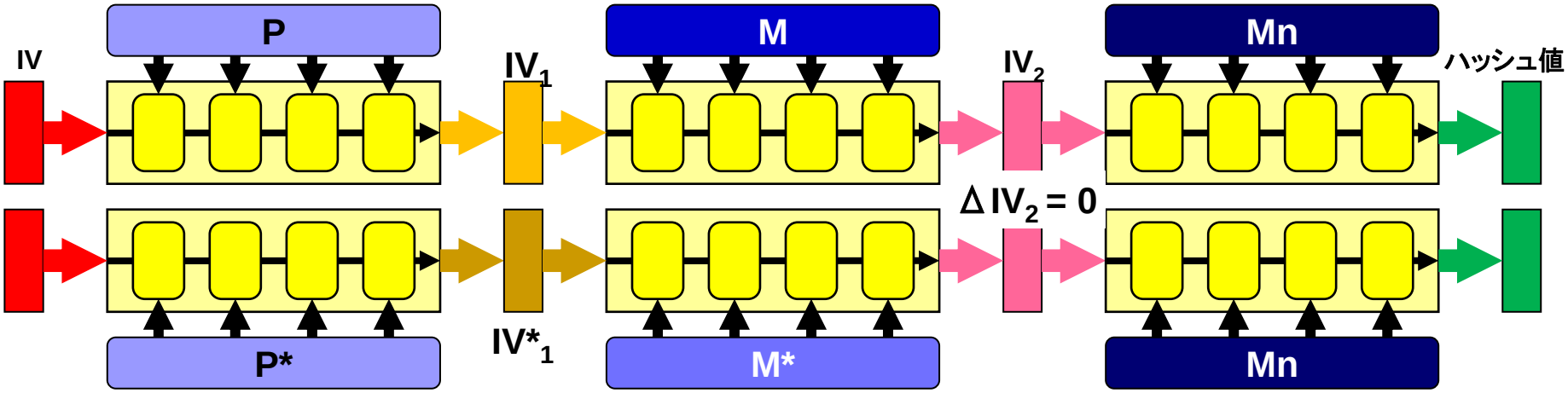
計算に使用されたPS3（研究者の一人であるBenne de Weger氏のWebサイトから引用）
[画像のクリックで拡大表示]

出典：日経パソコンより

MD5 (Merkle-Damgard構造) の衝突

- 衝突攻撃：2004年Wangらにより発見
  MD5 (M) =MD5 (M*) となる (M,M*) を見つける
- 選択プレフィックス衝突探索：2007年Stevensらが発見

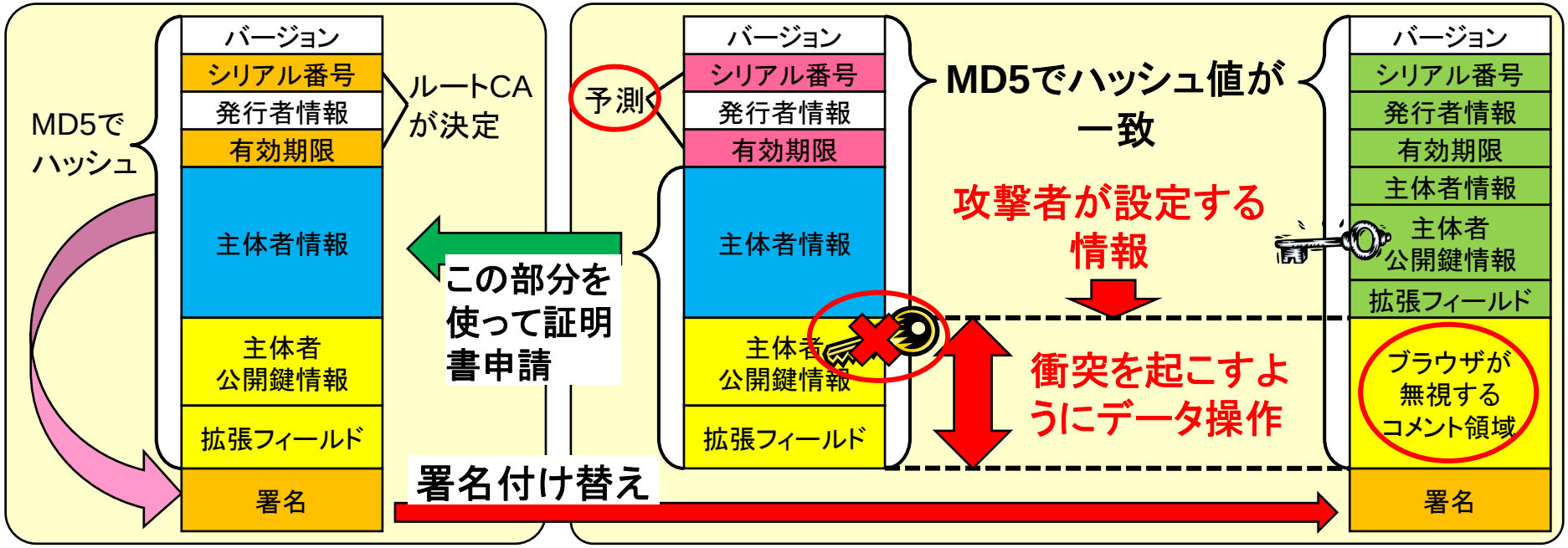
任意の (P,P*) に対しMD5 (P|M) =MD5 (P*|M*) となる (M,M*) を求める



- 2008年：PにRoot CAをだますためのダミー情報、P*に公開鍵を含む偽造情報を入れて選択プレフィックス衝突探索手法を実行し中間CA EE証明書を偽造。PS3 200台で約1日

MD5を使った偽造SSL証明書

- 何が問題だったのか？
  - MD5の耐衝突性が脆弱
  - ルートCAが決める「シリアルナンバー」「有効期限」が予測可
  - ブラウザが無視するコメント領域が存在



実際にどうなっているのか検証

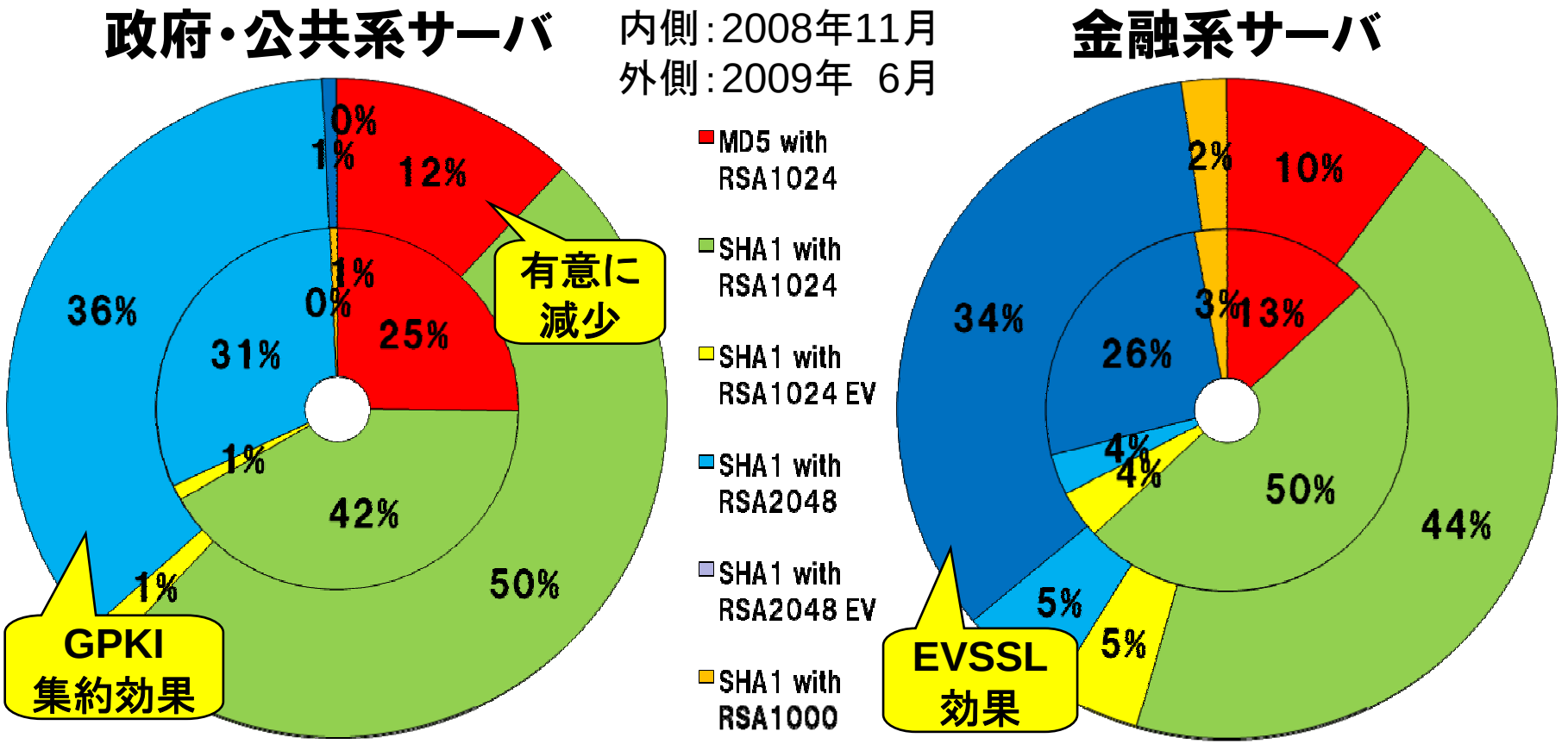
- **調査対象：**

  政府・公共系サイト及び金融系サイトの各トップページからたどることができるSSLサーバ

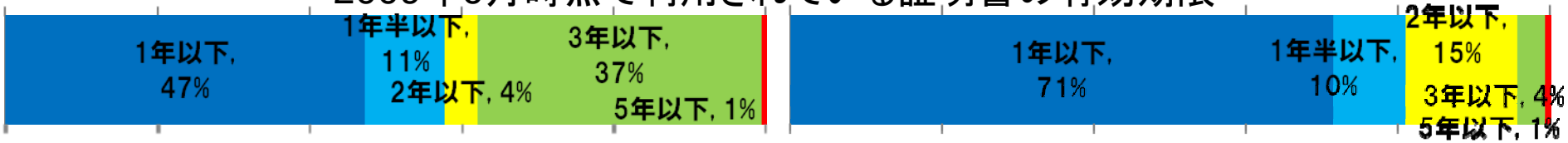
政府・公共系：約145サーバ、金融系：約135サーバ

- **調査期間：2008年10～11月／2009年5～6月**
- **調査内容：**
  - サーバ証明書の状況（有効期限、アルゴリズム、鍵長等）
  - 暗号選択設定の状況（接続可能なアルゴリズム）
  - ブラウザでの実際の接続状況（IE6, IE7, Firefox3）

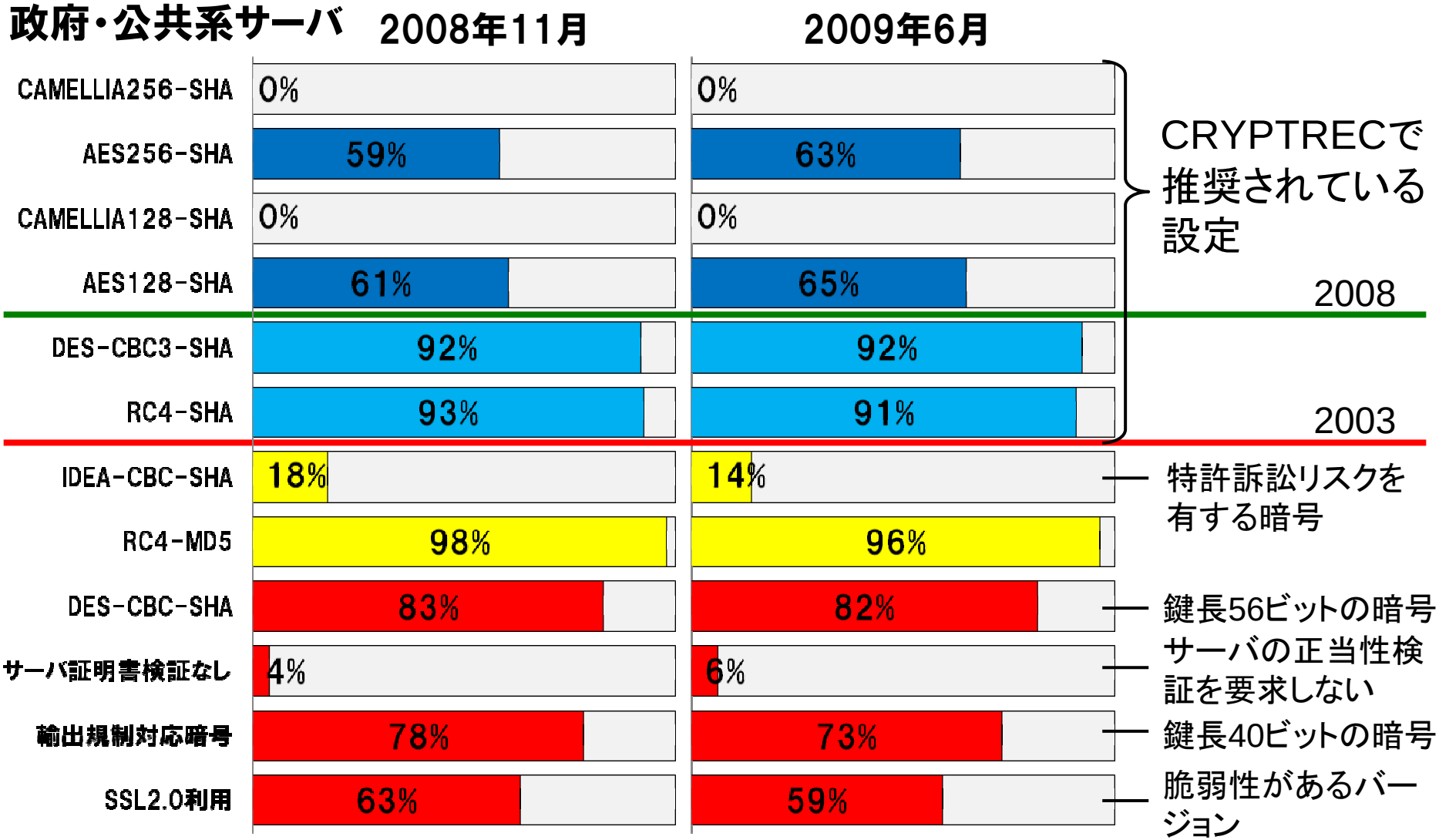
サーバ証明書の状況（アルゴリズムと有効期限）



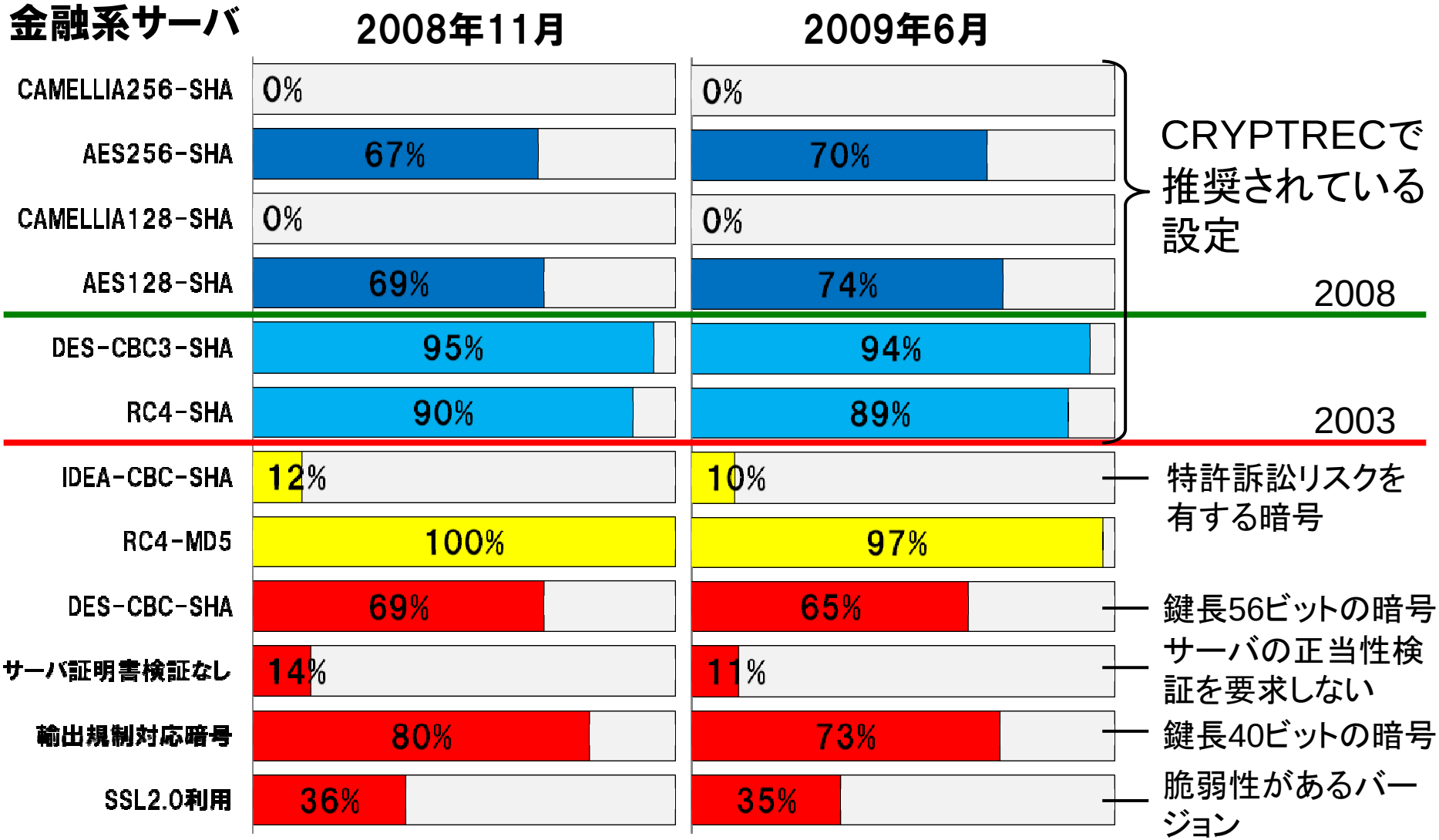
2009年6月時点で利用されている証明書の有効期限



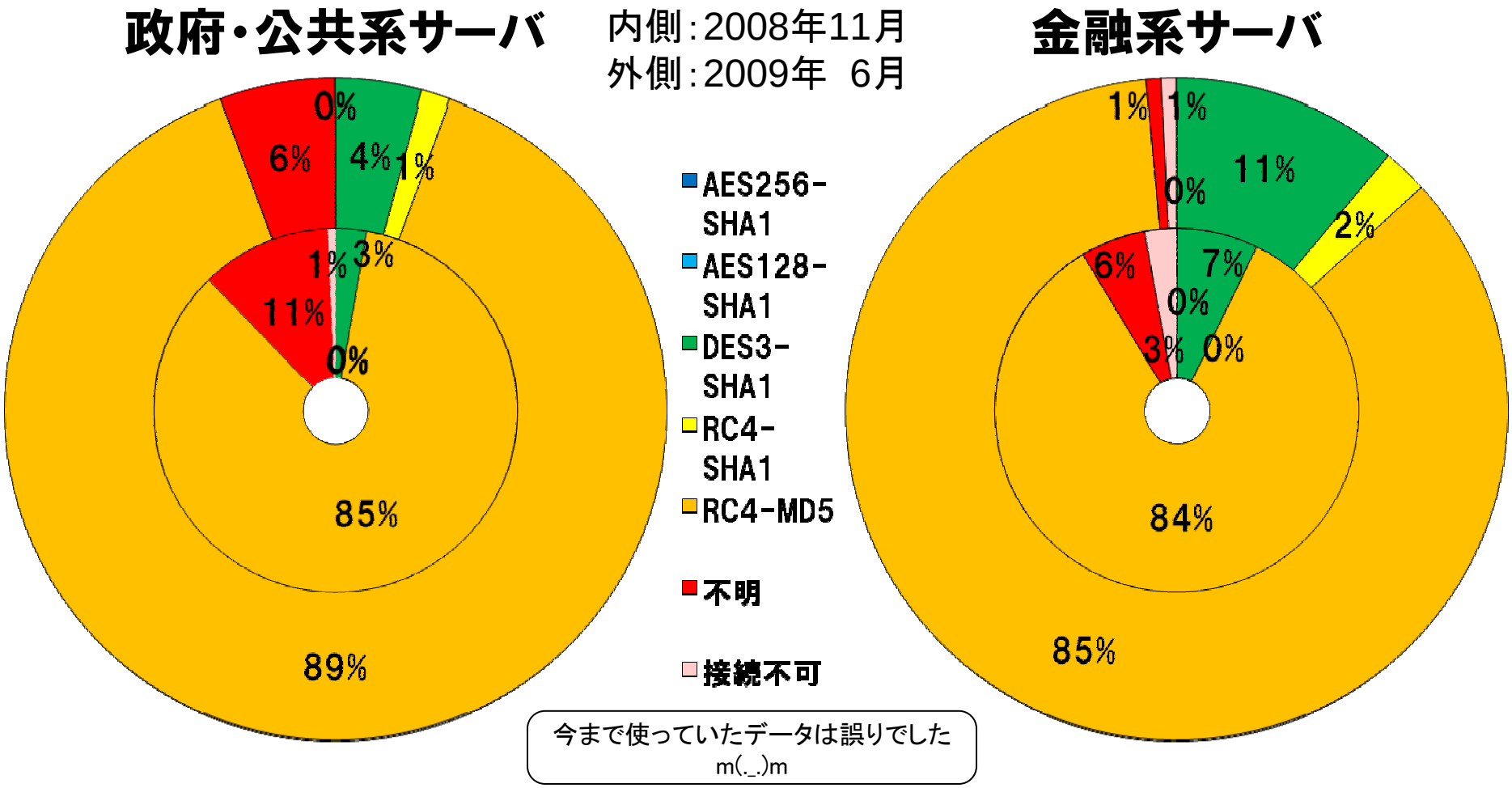
サーバでの暗号設定　～受入可能な主な暗号～



サーバでの暗号設定　～受入可能な主な暗号～

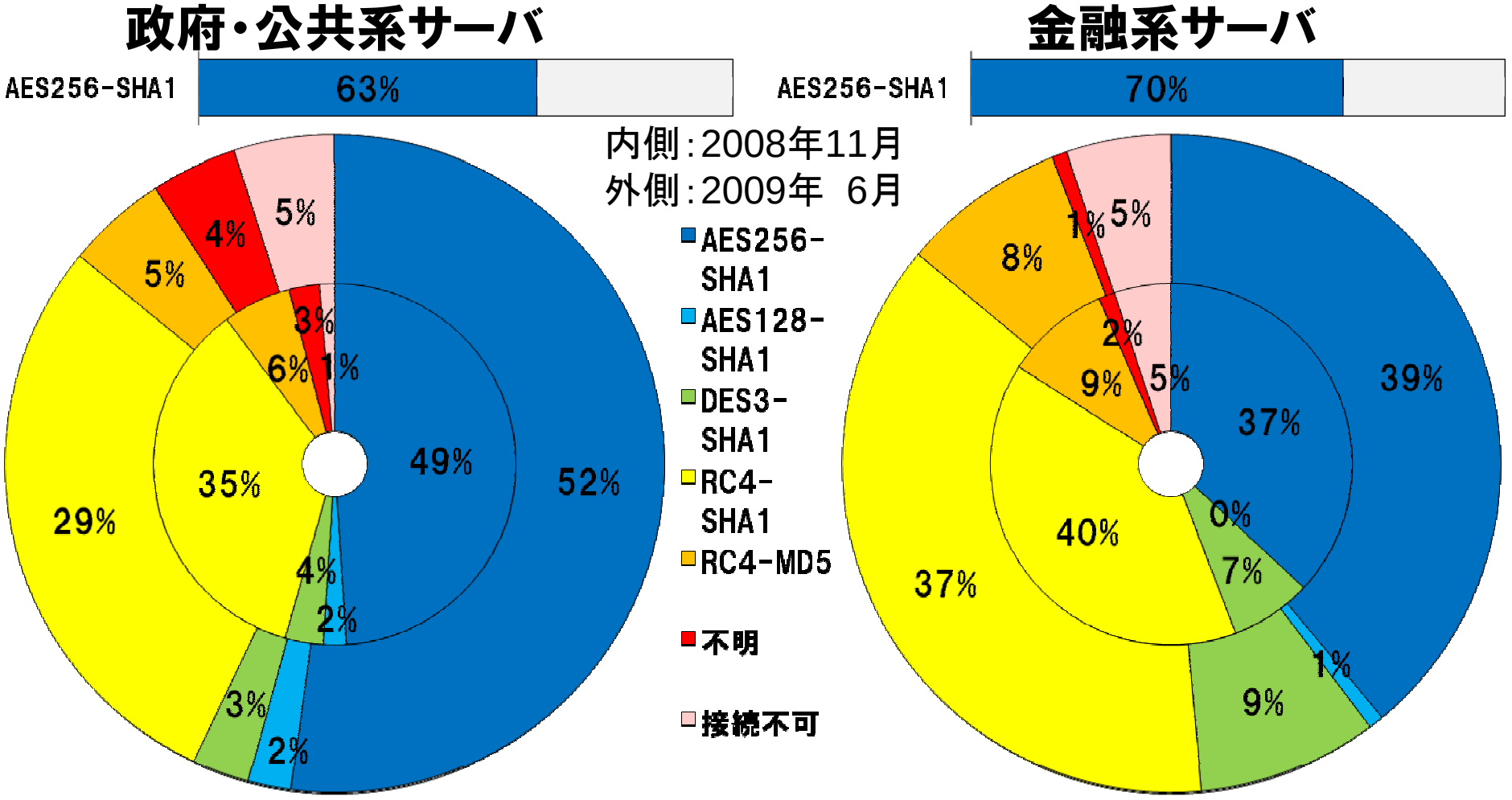


Internet Explorer 7 (Win XP)での接続



RC4はSHA1ではなくMD5でした。XP SP3でも同様

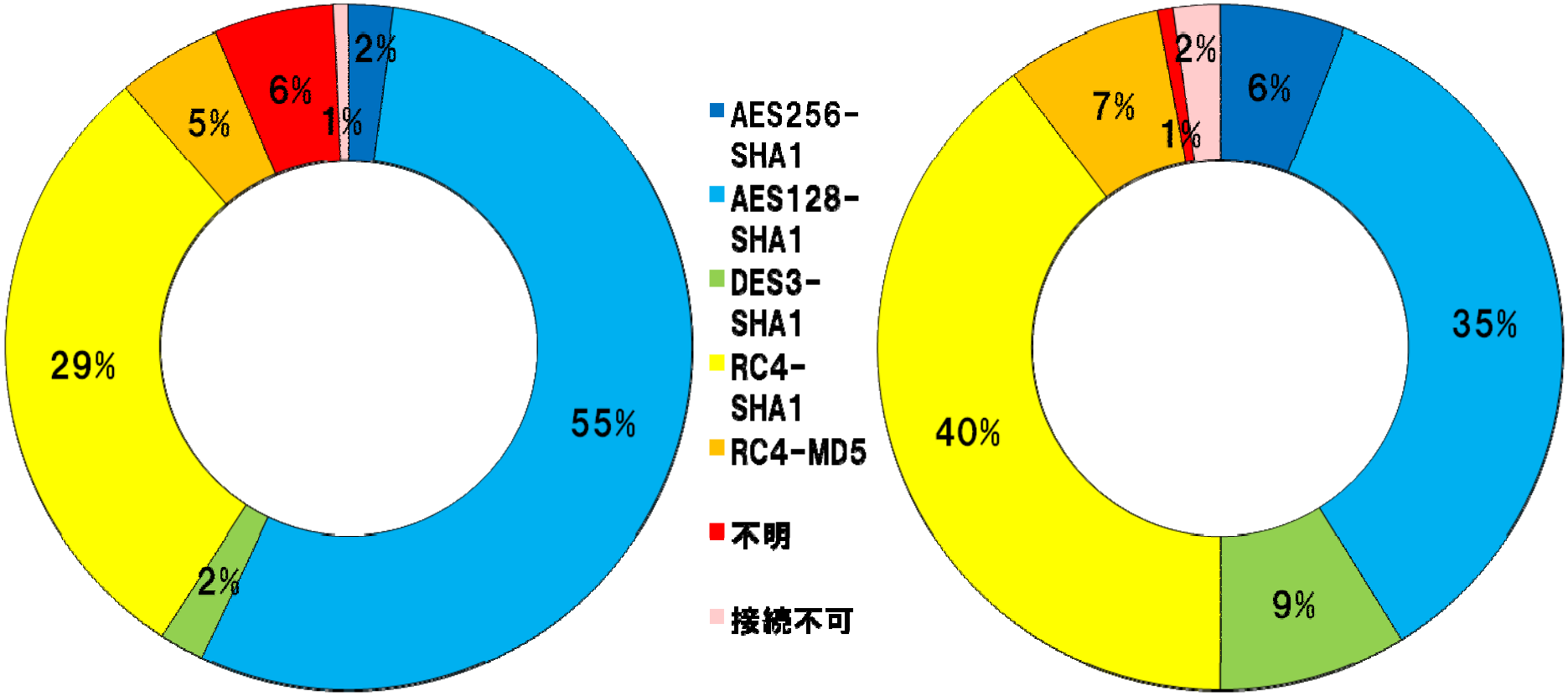
Firefox 3 (Win XP)での接続



AESの暗号設定状況と実際の接続での逆転現象

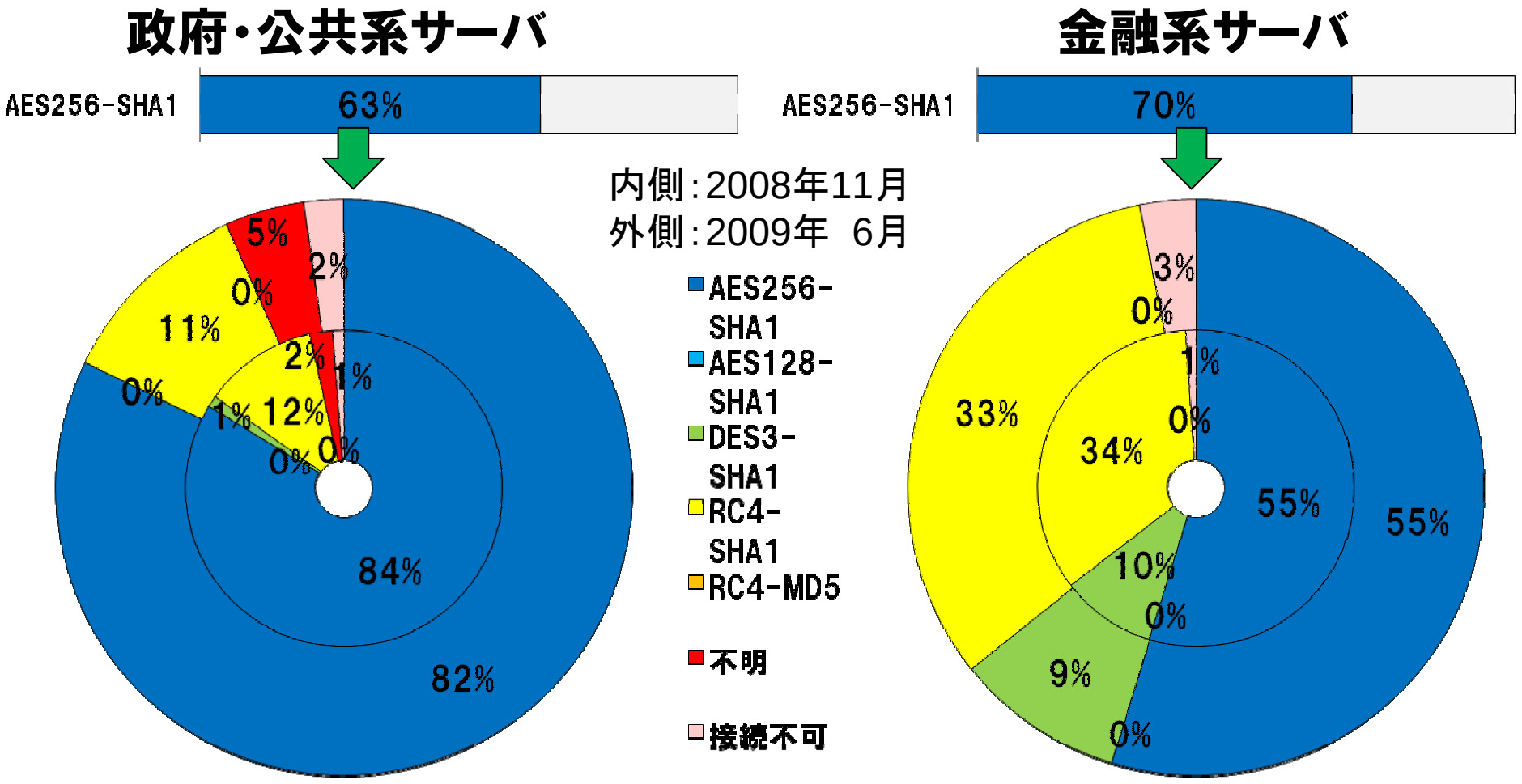
Internet Explorer 7 (Win Vista)での接続

政府・公共系サーバ　外側：2009年　6月　金融系サーバ



AESは使われるけども、AES128のほうが優先

AESが利用される設定になっているか



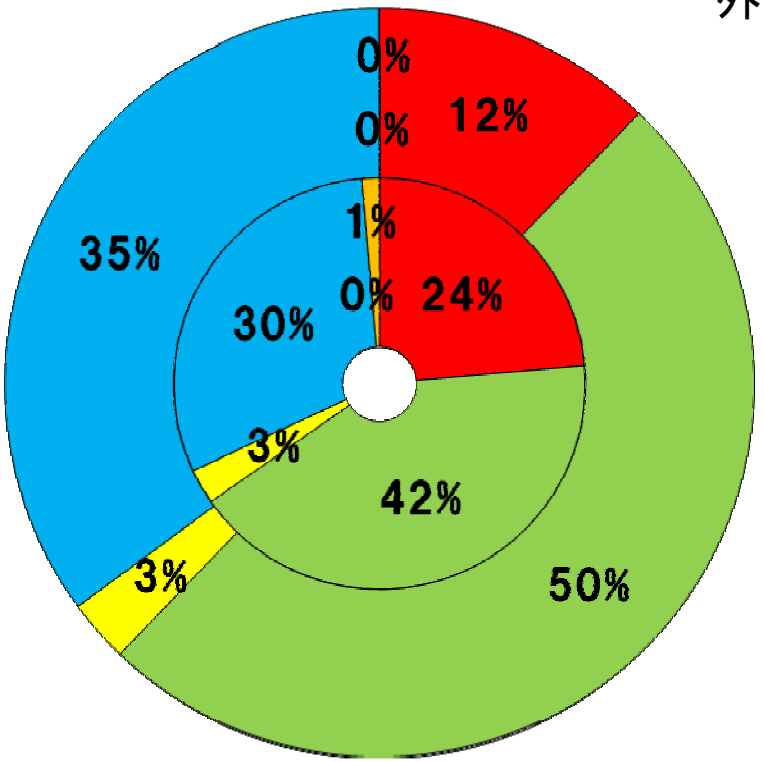
設定変更してもAESの利用率はほとんど変わっていない

SSLサーバ証明書と暗号のバランス

AES256-SHAで接続しているサーバに限定

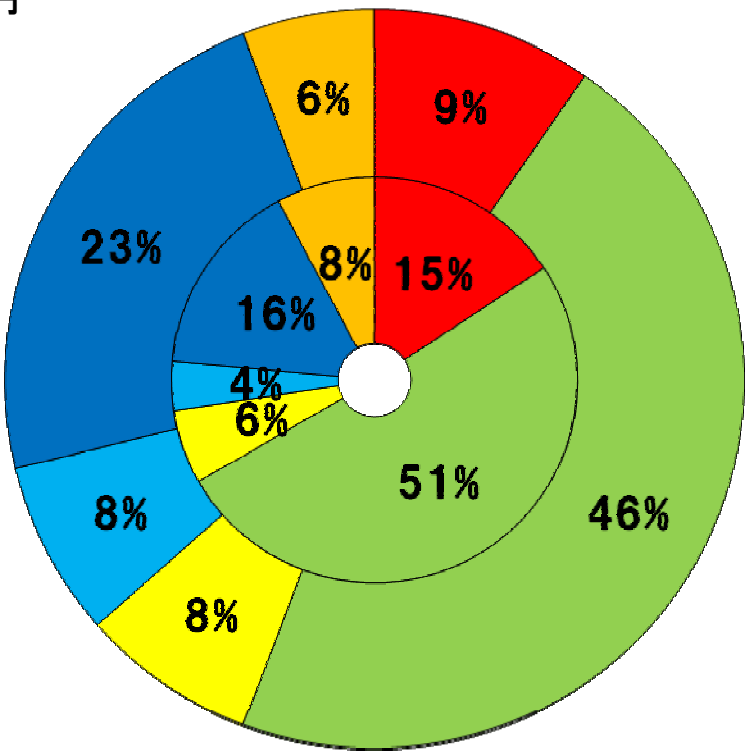
内側：2008年11月
外側：2009年　6月

政府・公共系サーバ



MD5 with RSA1024
SHA1 with RSA1024
SHA1 with RSA1024 EV
SHA1 with RSA2048
SHA1 with RSA2048 EV
SHA1 with RSA1000

金融系サーバ



AESを使っているサーバだから証明書も、とはなっていない

SSLの調査結果から

整合的に暗号設定が行われている形跡が見いだせない
～ 暗号アルゴリズム設定がベンダ任せになっていませんか ～

- 「暗号化はAES256ビット」なのに「鍵交換は512ビットRSA」
- RC4で接続といってもSHA1を使っているとは限らない
- AESを使えるようにしたのにMD5のサーバ証明書のまま
- 同じ企業内のサーバであっても設定内容に一貫性がない
- どの暗号アルゴリズムが選択されたかブラウザ(クライアント)からは確認できない

せっかく設定を変えたのに・・・設定失敗

- AESを使えるようにしたらSSL2.0やEXPも利用可能になった
- 「AES256-SHA1」が使えるはずなのに「RC4-MD5」を選択

SSLの調査結果から

SSL証明書の中身を確認しているか？

- 更改できるタイミングが何回もあったはずなのにMD5のSSL証明書が未だに健在
- MD5や署名長は変わっても、鍵交換用RSAの鍵長は変わらない

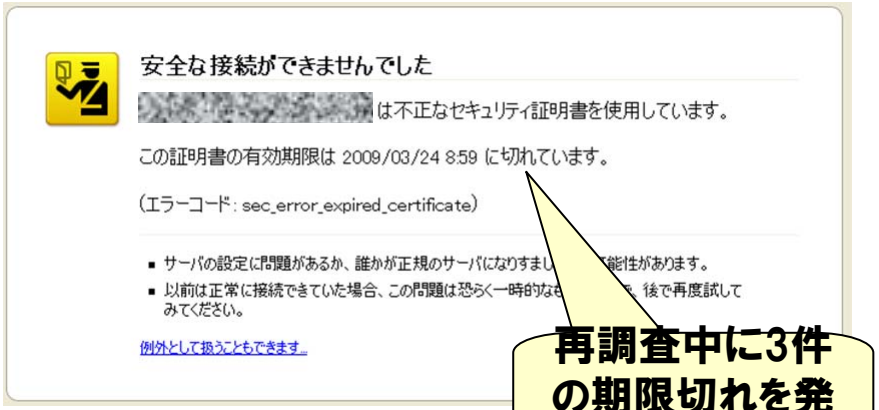
SSL証明書の期限切れを起こすことが意外と多い

全日空のCIO、搭乗システム障害について会見、「担当者の会話が不十分だったためのごく初歩的なミス」と反省の弁

全日空幹部は2008年9月18日会見を開き、14日に発生したシステム不具合の原因を公表し反省を語った。払い戻しなど直接的な損失額は、全日空グループ全体で2億円。

原因は、既報されているように、チェックイン端末を管理するサーバー内の暗号化機能の有効期限の設定ミスによるもの。今回のトラブルについて、同社のCIO(最高情報責任者)である上席執行役員の佐藤透IT推進室長は、2点を挙げた。

記者会見の冒頭謝罪する全日空の経営陣
[画像のクリックで拡大表示]



再調査中に3件の期限切れを発見

出典：日経情報ストラテジー
http://itpro.nikkeibp.co.jp/article/NEWS/20080918/315052/

まとめ

SSLサーバの暗号設定マニュアルのようなものを整備
～ 暗号の説明というより作業マニュアルとして反映 ～

- 利用するciphersuiteを設定するための作業マニュアル
- SSLサーバ証明書を発行・更新手続きするための作業マニュアル
- 作業マニュアルを利用させるためのルール化

例えば NIST SP800-52

NIST Special Publication 800-52

Guidelines for the Selection and Use of Transport Layer Security (TLS) Implementations

NIST
National Institute of Standards and Technology
Technology Administration
U.S. Department of Commerce

C. Michael Chernick, Charles Edington III, Matthew J. Fanto, Rob Rosenthal

COMPUTER SECURITY

5.1 Considerations for Selecting TLS Client Implementations

Clients play a limited, but crucial role in the overall security posture. The client negotiates three parameters: the protocol version, the cipher suite, and the compression algorithm. These items are presented in the "ClientHello" message and form the basis for the server to negotiate the strongest possible security options.

The ClientHello message is the first message to be sent as the client establishes a TLS connection to the server. These messages allow the client to stay connected, re-establish an existing session, or to establish several independent secure sessions without repeating the full handshake procedure.

The client version field within the ClientHello message represents the protocol version that client supports. This field should contain the highest version number the client is prepared to support. For implementations that support TLS this value is: major=3, minor=1 (which represents 3.1, and hence TLS). All non-TLS implementations should use major=3, minor=0 for SSLv3. This designation does not limit the implementation to the identified protocol version. For example, if a client wishes to use only TLS, the client must connect to the server and is responsible for terminating the connection if the server selects any other protocol. Under no circumstances should a client use any protocol less than SSLv3[20]. For the most secure protection of data, only use clients that support TLS and that can disable all versions of SSL.

Table 2: Recommended Client Cipher Suites[21]

Cipher Suite	Authent-ication	Key Establishment	Encryption	Digest
TLS_DHE_DSS_WITH_AES_256_CBC_SHA	DSS	DHE	AES_256_CBC	SHA-1
TLS_DHE_RSA_WITH_AES_256_CBC_SHA	RSA	DHE	AES_256_CBC	SHA-1
TLS_RSA_WITH_AES_256_CBC_SHA	RSA	RSA	AES_256_CBC	SHA-1
TLS_DH_DSS_WITH_AES_256_CBC_SHA	DSS	DH	AES_256_CBC	SHA-1
TLS_DH_RSA_WITH_AES_256_CBC_SHA	RSA	DH	AES_256_CBC	SHA-1
TLS_DHE_DSS_WITH_AES_128_CBC_SHA	DSS	DHE	AES_128_CBC	SHA-1
TLS_DHE_RSA_WITH_AES_128_CBC_SHA	RSA	DHE	AES_128_CBC	SHA-1
TLS_RSA_WITH_AES_128_CBC_SHA	RSA	RSA	AES_128_CBC	SHA-1
TLS_DH_DSS_WITH_AES_128_CBC_SHA	DSS	DH	AES_128_CBC	SHA-1
TLS_DH_RSA_WITH_AES_128_CBC_SHA	RSA	DH	AES_128_CBC	SHA-1
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA	DSS	DHE	3DES_EDE_CBC	SHA-1
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA	RSA	DHE	3DES_EDE_CBC	SHA-1
TLS_RSA_WITH_3DES_EDE_CBC_SHA	RSA	RSA	3DES_EDE_CBC	SHA-1
TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA	DSS	DH	3DES_EDE_CBC	SHA-1
TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA	RSA	DH	3DES_EDE_CBC	SHA-1
TLS_RSA_WITH_RC4_128_SHA[22]	RSA	RSA	RC4_128	SHA-1

Table 3: Recommended Server Cipher Suites[24]

Cipher Suite	Auth	Key Establishment	Encryption	Digest
TLS_DHE_DSS_WITH_AES_256_CBC_SHA	DSS	DHE	AES_256_CBC	SHA-1
TLS_DHE_RSA_WITH_AES_256_CBC_SHA	RSA	DHE	AES_256_CBC	SHA-1
TLS_RSA_WITH_AES_256_CBC_SHA	RSA	RSA	AES_256_CBC	SHA-1
TLS_DH_DSS_WITH_AES_256_CBC_SHA	DSS	DH	AES_256_CBC	SHA-1
TLS_DH_RSA_WITH_AES_256_CBC_SHA	RSA	DH	AES_256_CBC	SHA-1
TLS_DHE_DSS_WITH_AES_128_CBC_SHA	DSS	DHE	AES_128_CBC	SHA-1
TLS_DHE_RSA_WITH_AES_128_CBC_SHA	RSA	DHE	AES_128_CBC	SHA-1
TLS_RSA_WITH_AES_128_CBC_SHA	RSA	RSA	AES_128_CBC	SHA-1
TLS_DH_DSS_WITH_AES_128_CBC_SHA	DSS	DH	AES_128_CBC	SHA-1
TLS_DH_RSA_WITH_AES_128_CBC_SHA	RSA	DH	AES_128_CBC	SHA-1
TLS_DHE_DSS_WITH_AES_256_CBC_SHA	DSS	DHE	AES_256_CBC	SHA-1
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA	DSS	DHE	3DES_EDE_CBC	SHA-1
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA	RSA	DHE	3DES_EDE_CBC	SHA-1
TLS_RSA_WITH_3DES_EDE_CBC_SHA	RSA	RSA	3DES_EDE_CBC	SHA-1
TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA	DSS	DH	3DES_EDE_CBC	SHA-1
TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA	RSA	DH	3DES_EDE_CBC	SHA-1

Note that all server certificates with RSA keys should have a key length of at least 1024 bits.

例えば NIST SP800-44

NIST Special Publication 800-44
Version 2

Guidelines on Securing Public Web Servers

Recommendations of the National Institute of Standards and Technology

Miles Tracy, Wayne Jansen, Karen Scarfone, and Theodore Winograd

C O M P U T E R S E C U R I T Y

7.7 Checklist for Using Authentication and Encryption Technologies for Web Servers

Completed	Action
	Configure Web authentication and encryption technologies
☐	For Web resources that require minimal protection and for which there is a small, clearly defined audience, configure address-based authentication
☐	For Web resources that require additional protection but for which there is a small, clearly defined audience, configure address-based authentication as a second line of defense
☐	For Web resources that require minimal protection but for which there is no clearly defined audience, configure basic or digest authentication (better)
☐	For Web resources that require protection from malicious bots, configure basic or digest authentication (better) or implement mitigation techniques discussed in Section 5.2.4
☐	For organizations required to comply with FIPS 140-2, ensure the SSL/TLS implementation is FIPS-validated
☐	For Web resources that require maximum protection, configure SSL/TLS
	Configure SSL/TLS
☐	Ensure the SSL/TLS implementation is fully patched
☐	Use a third-party issued certificate for server authentication (unless all systems using the server are organization-managed, in which case a self-signed certificate could potentially be used instead)
☐	For configurations that require a medium level of client authentication, configure server to require username and password via SSL/TLS
☐	For configurations that require a high level of client authentication, configure server to require client certificates via SSL/TLS
☐	Ensure weak cipher suites are disabled (see Table 7.1 for the recommended usage of Federal cipher suites)
☐	Configure file integrity checker to monitor Web server certificate
☐	If only SSL/TLS is to be used in the Web server, ensure access via any TCP port other than 443 is disabled
☐	If most traffic to the Web server will be via encrypted SSL/TLS, ensure that appropriate logging and detection mechanisms are employed in the Web server (because network monitoring is ineffective against encrypted SSL/TLS sessions)
	Protect against brute force attacks
☐	Use strong authentication if possible
☐	Use a delay after failed login attempts
☐	Lock out an account after a set number of failed login attempts
☐	Enforce a password policy
☐	Blacklist IP addresses or domains known to attempt brute force attacks
☐	Use log monitoring software to detect brute force attacks

☐	Ensure weak cipher suites are disabled (see Table 7.1 for the recommended usage of Federal cipher suites)

CRYPTREC推奨

SSL/TLSに対する(たぶん国内唯一の)公式ガイドライン

2003年2月決定の電子政府推奨暗号リストに基づく推奨

公開鍵暗号（署名）	RSA 1024 bit以上 DSA 1024 bit以上 ECDSA 160 bit 以上	SHA-1
公開鍵暗号（鍵共有）	RSA 1024 bit以上 DH 1024 bit以上 ECDH 160 bit以上	SHA-1
共通鍵暗号	AES 128 bit以上 Camellia 128 bit以上 3-key Triple DES 128-bit RC4	

➡ MD5はこの時点ですでに推奨ではなかった

2008年3月発行の電子政府推奨暗号の利用方法に関するガイドブックに記載の推奨

公開鍵暗号（署名）	RSA 2048 bit以上 DSA 2048 bit以上 ECDSA 224 bit 以上	SHA-1*
公開鍵暗号（鍵共有）	RSA 2048 bit以上 DH 2048 bit以上 ECDH 192 bit以上	SHA-1*
共通鍵暗号	AES 128 bit以上 Camellia 128 bit以上	

* 利用は推奨されないが、変更できるようになった場合には暗号切替等を検討することを推奨

➡ RC4とTriple DESも推奨から外されている

カナダ政府の推奨

Communications Security Establishment Canada | Centre de la sécurité des télécommunications Canada

IT Security Bulletin
Bulletin de sécurité TI

November 2008 | ITSB-60 | novembre 2008

Guidance on the Use of the Transport Layer Security Protocol within the Government of Canada

Purpose

The purpose of this Bulletin is to provide Government of Canada (GC) departments guidance on:

- using the Transport Layer Security (TLS) protocol for the protection of Protected A and B information;
- the approved cryptographic protocols and algorithms that the Communications Security Establishment Canada (CSEC) recommends for use with TLS; and
- standards and NIST special publications that describe the recommended cryptographic primitives and provide additional information on TLS.

Conseils
TLS (*Tran*
du gouve

Objet

Le présent bu
gouvernemen

- l'utilisatic transport (pour la pr A et PRO
- les protoc approuvés télécomm d'utiliser
- les norme décrivent recommar additionn

Table 1 : Approved Cryptographic Primitives for TLS/
Tableau 1 : Primitives cryptographiques approuvées pour le protocole TLS

Key Establishment/ Établissement de clés	Block Ciphers/ Chiffrement par bloc	Hash Functions/ Fonctions de hachage	Digital Signatures/ Signatures numériques	Random Bit Generation/ Génération de bits aléatoires	Integrity Protection/ Protection de l'intégrité
RSA	AES	SHA-1	DSA	Hash_DRBG	HMAC
Diffie-Hellman	Triple DES	SHA-224	RSA	HMAC_DRBG	CMAC
Key Exchange Algorithm (KEA)	CAST5	SHA-256	Elliptic Curve DSA	CTR_DRBG	
Elliptic Curve Diffie-Hellman	SKIPJACK	SHA-384		Dual_EC_DRBG	
Elliptic Curve MQV		SHA-512		Legacy DRBGs based on DES, Triple DES, AES, SHA-1, and HMAC / Anciens générateurs de bits aléatoires déterministes fondés sur DES, Triple DES, AES, SHA-1, et HMAC.	