INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN

Flameと暗号技術の怪しい関係・・・
マルウェアの片棒を担いだ暗号技術
― まさかの“マイクロソフト製”マルウェア！？

IPA　技術本部　セキュリティセンター
暗号グループ
神田 雅透

去年の沖縄ICTフォーラムで

Comodo Hackerって何者？

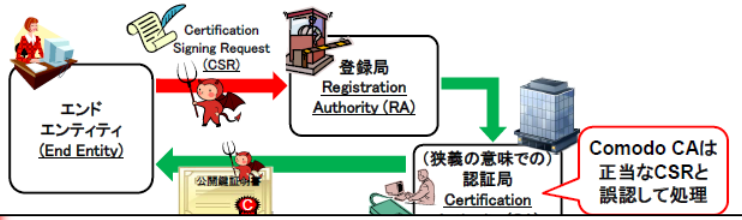
Comodo, DigiNotarを始め、複数のCAを攻撃したとの犯行声明を出したComodo Hacker

- 「イラン在住の21歳の一匹狼のクラッカー」と自称
- 「イラン政府や軍とは無関係」と主張
- 「イラン反体制派組織に恐怖を与え、イラン国民、核技術者、大統領の守護者」を自任
- 同一人物の攻撃であることの痕跡をあえて残している

2011

Comodoのケース

- 不正SSLサーバ証明書が通常の手続きに則って発行
  - Comodo RAの審査を不正にすり抜けた結果、見掛け上正当な偽CSRに基づいて不正SSLサーバ証明書を正規発行
    - 2011年3月15日、Comodo RAに存在するユーザアカウントをクラック（主にイランに割り当てられているIPアドレスが使われた）
    - クラックされたユーザアカウント上に新たなユーザIDを作る
    - 新たなユーザIDで見掛け上正当なCSRを9つ（7ドメイン）不正に作る



2011

Comodo Hackerって何者？

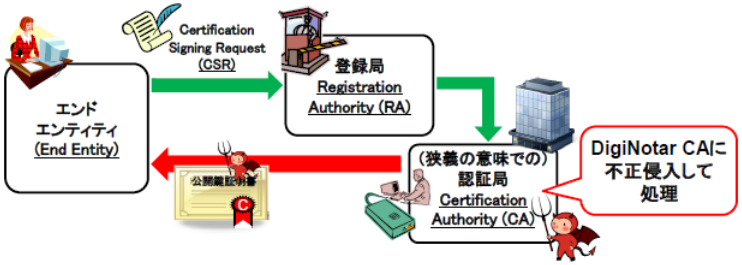
米国などが主導するようなインターネット社会や情報化社会を否定、IT基盤に打撃を与えることが目的

- イラン核問題をはじめとする、イラン政府やイラン国民に対する米国やイスラエルの攻撃に対する報復を示唆
- DigiNotarを狙ったのはオランダへの報復と主張
  - オランダGPKIに打撃を与える目的
- 少なくともさらに3つ以上のCAに対して攻撃が成功？
  - StartCOM(本拠：イスラエル)：
    - HSM接続成功、電子メールやDBバックアップ、顧客情報などを入手
  - GlobalSign(本拠：日本)：
    - 全サーバへのアクセス成功、DBバックアップ、 ならびに米国のglobalsign.comドメインの個別鍵を入手
    - 発行業務一時停止を含む緊急対応により安全性を確認

20111215沖縄ICTフォーラム2011 13

DigiNotarのケース

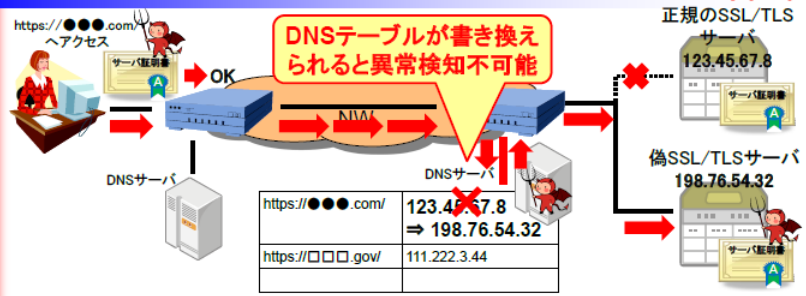
- 不正SSLサーバ証明書がCA機能を乗っ取られて発行
  - EV-SSLサーバ証明書発行用CAを含め、少なくとも6つのCA（疑いを含めると30個のCA）に不正侵入され、不正SSLサーバ証明書を発行
    - 2011年7月19日に128枚、20日に129枚発行されたのを含め、少なくとも合計531枚の不正SSLサーバ証明書が発行されていた
    - 2011年6月17日から今回の攻撃が始まっていたことを把握



20111215沖縄ICTフォーラム2011 14

何が問題視されたのか

実害が発生した可能性が高い



「政府機関(体制側)等による盗聴行為」が イラン国内で実際に行われた可能性がある

- イラン周辺で不正発行されたSSLサーバ証明書に対するOCSPリクエストが多発
- 不正発行されたSSLサーバ証明書に、Googleのほか

PKIの危機を招いたもの

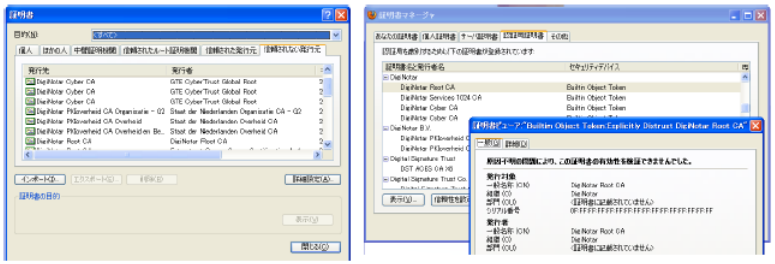
ルートCAのずさんな運営管理と見過ごした監査体制 ～ ルートCAの水準と監査品質の均一性への懸念 ～

- ルートCAとしてはあまりにも重大な失態が相次ぐ
  - 事件報道されるまでの5週間、事実を隠ぺいし続けた
    - 2011年7月19日以降、短期間に不正SSLサーバ証明書の発行・失効処理が繰り返されていたにも関わらず、根本的な対策を取らなかった
    - 2011年6月17日から今回の攻撃が始まっていたことを把握
    - 7月28日イランで不正SSLサーバ証明書が悪用されていることを把握
    - OSやブラウザ等のベンダにもその事実を通知しなかった

不正*.google.com証明書のOCSPリクエスト

PKIの危機を招いたもの

- 主要ブラウザベンダの対処
  - 事件報道後、主要ブラウザベンダは緊急の修正パッチを提供
    - 対策:**DigiNotarのルート証明書を削除**

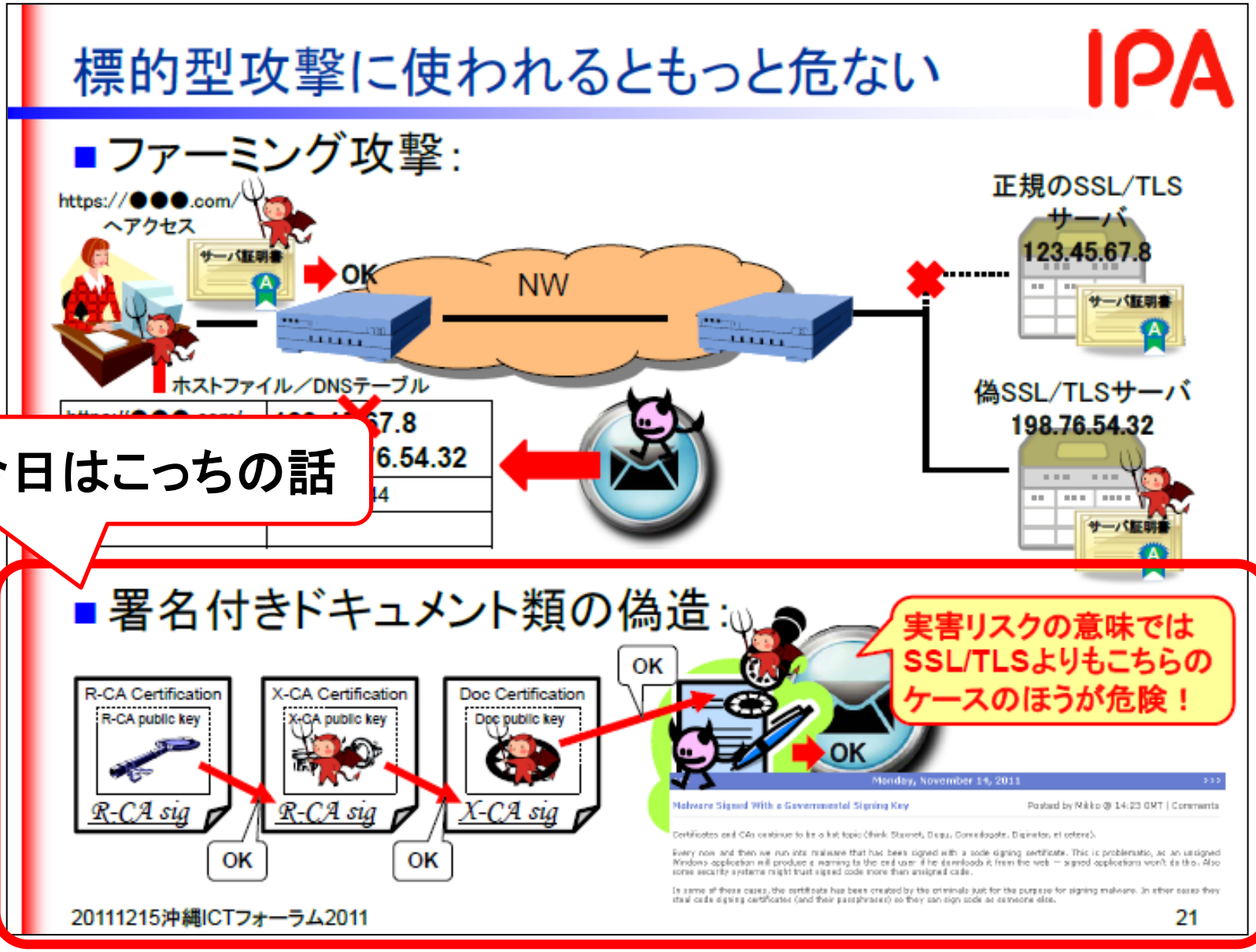


➡ DigiNotarの業務停止
破産手続き開始

オランダGPKIのルートCAの一つが潰れた
⇒ Comodo Hackerの目的達成


VASCO Announces Bankruptcy Filing by DigiNotar B.V.

もう一つ大事なことをいっていた



公開鍵証明書が悪用されるのはどんな時？

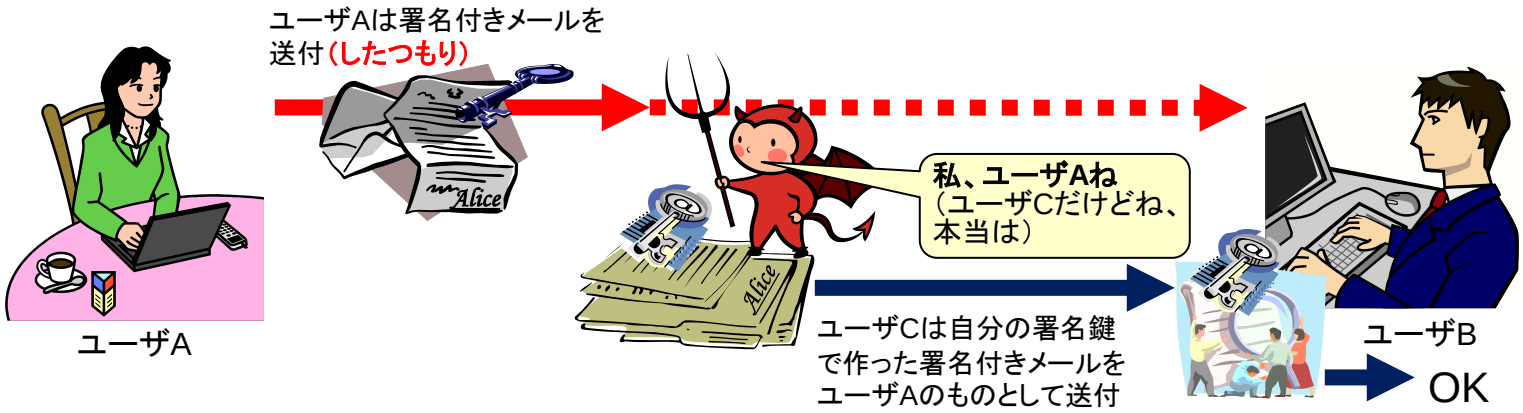
- 公開鍵証明書に対応する**秘密鍵が流出**（例：マレーシア政府の署名鍵流出事件）
- **登録局での検証ミス**によって不正な公開鍵証明書を認証局が誤って発行（例：Comodo事件）
- **認証局への不正アクセス**によって不正な公開鍵証明書を発行（例：Diginotar事件）
- 公開鍵証明書で使っている暗号技術が弱いことを突いて不正な公開鍵証明書を偽造
  - 公開鍵情報から**秘密鍵を割り出す**（公開鍵暗号の問題）
  - **真正な公開鍵証明書と区別ができない不正な公開鍵証明書を計算機によって偽造（ハッシュ関数の問題）**

で、本題に入る前に・・・

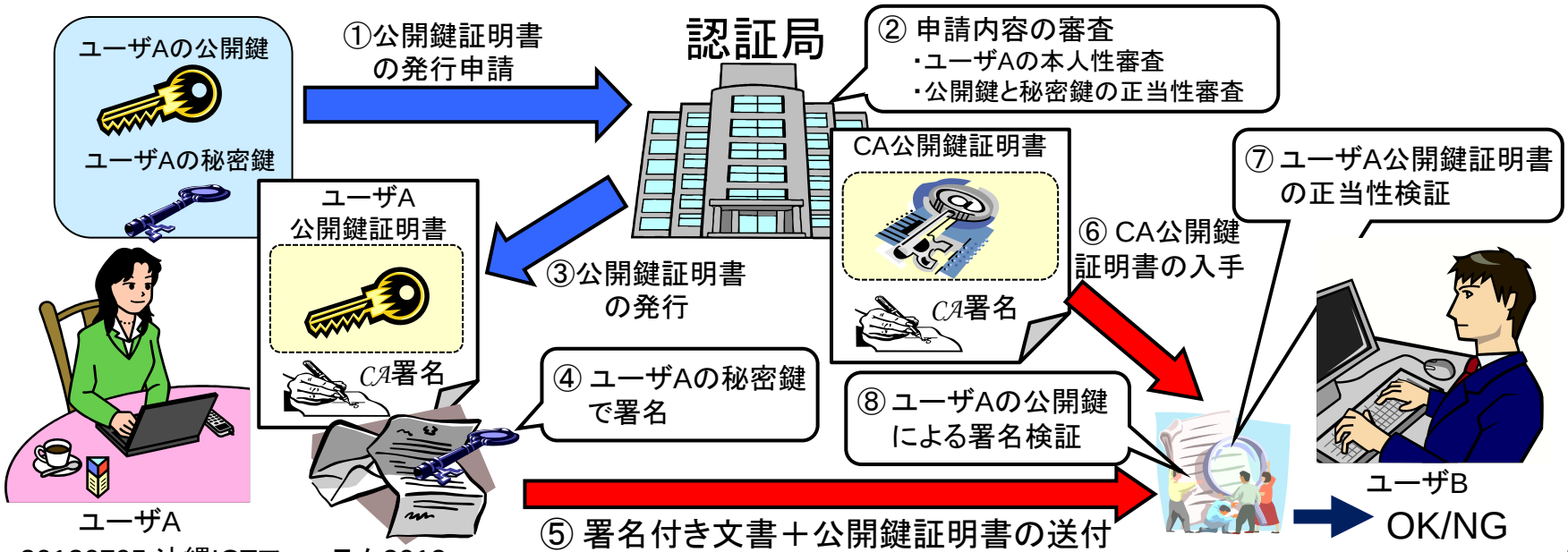
いくつかの準備を

公開鍵証明書の役目

■ 署名の検証鍵は本当に意図した相手のものか？

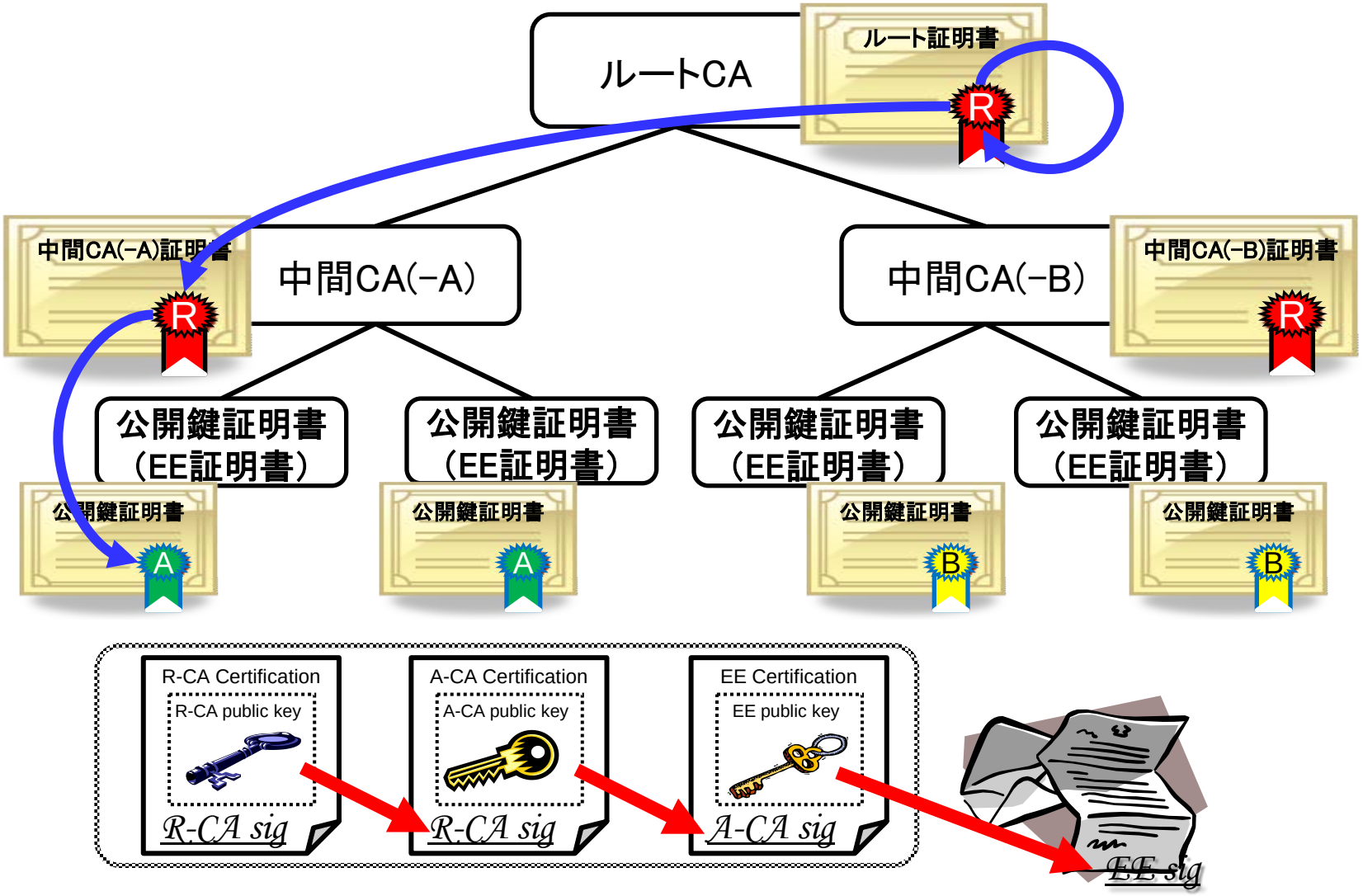


■ 公開鍵証明書は騙されないための保険



公開鍵基盤PKI (Public Key Infrastructure)

- ルートCAはPKIのTrust Anchor

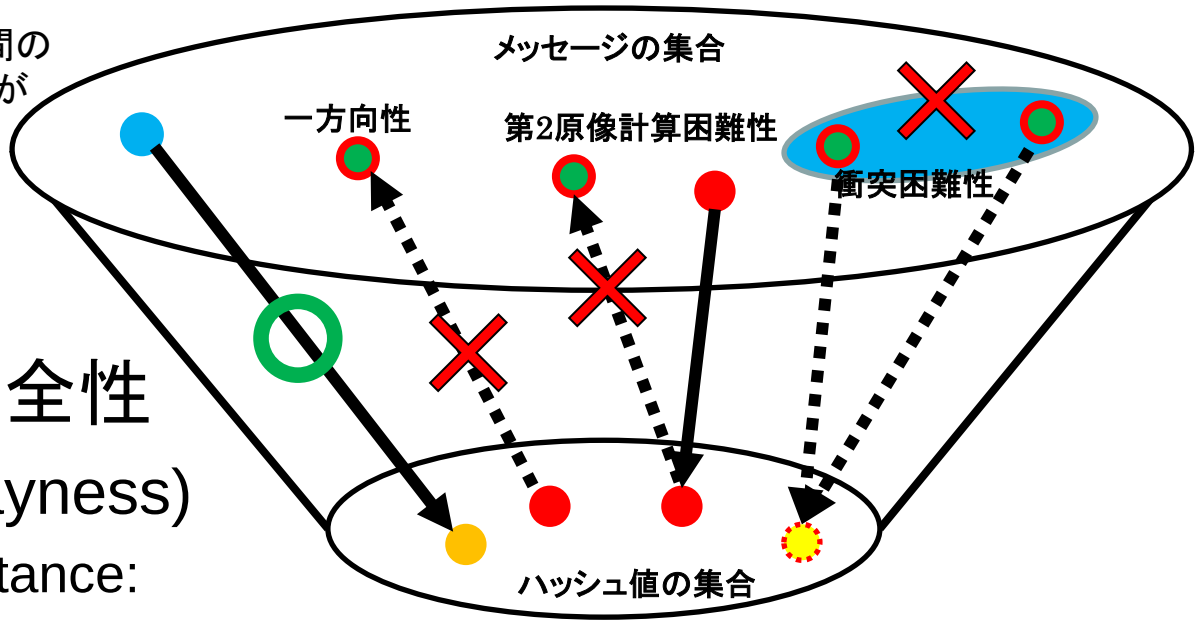


ハッシュ関数の性質

ハッシュ関数

任意長のメッセージを一定長のダイジェストに写像

メッセージ空間よりもダイジェスト空間のほうが小さいので、理論上必ず衝突が起こる（この点が暗号と異なる）



ハッシュ関数の安全性

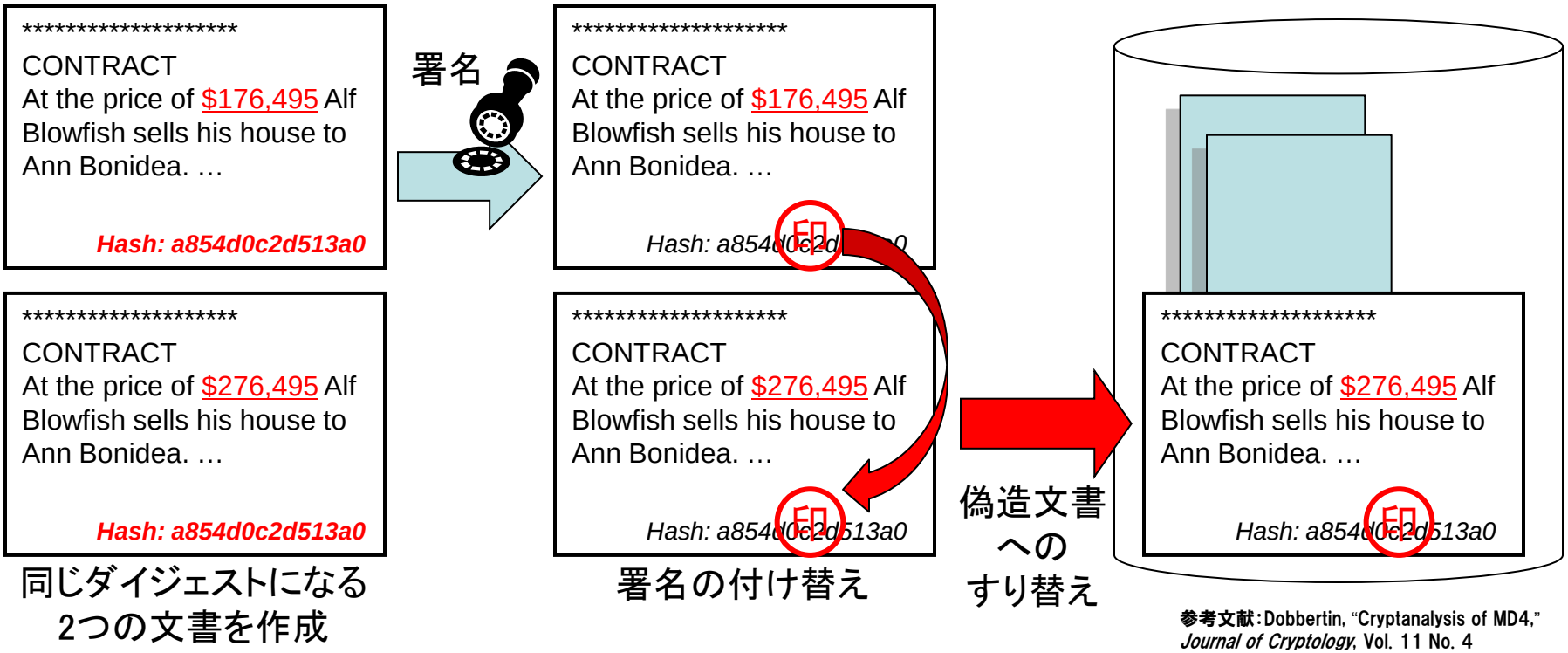
- 一方向性(Onewayness)
  - Preimage Resistance:
    ダイジェストからもとのメッセージ（の候補）が導き出せない
  - Second Preimage Resistance:
    特定のダイジェストに一致する別メッセージを作り出せない
- **衝突困難性・非衝突性(Collision-free/resistance)**
  - **同じダイジェストになる異なるメッセージ組を作り出せない**

ハッシュ衝突攻撃ができると何が起こるか

例えば、デジタル署名のすり替え（改竄）が可能になる

- 改竄検知が不可能となる文書を**あらかじめ**用意可能
  - X.509の偽造証明書もこのケース

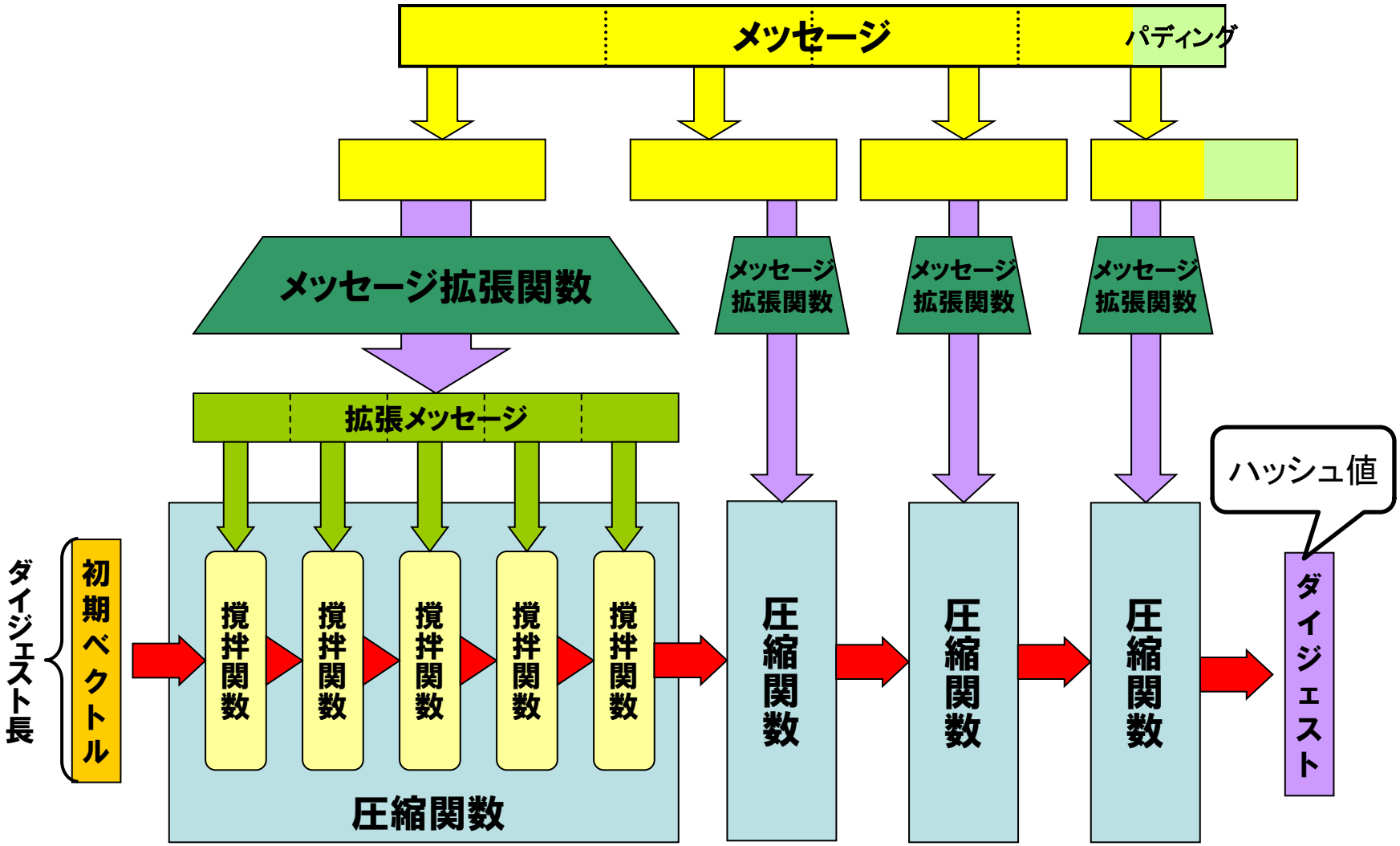
概念的にはこのケースが現実化したということ



参考文献：Dobbertin, “Cryptanalysis of MD4,” *Journal of Cryptology*, Vol. 11 No. 4

Merkle-Damgård構造　～一般的な構造～

MD4, MD5, SHA-1, SHA-2 は全てこの構造を採用

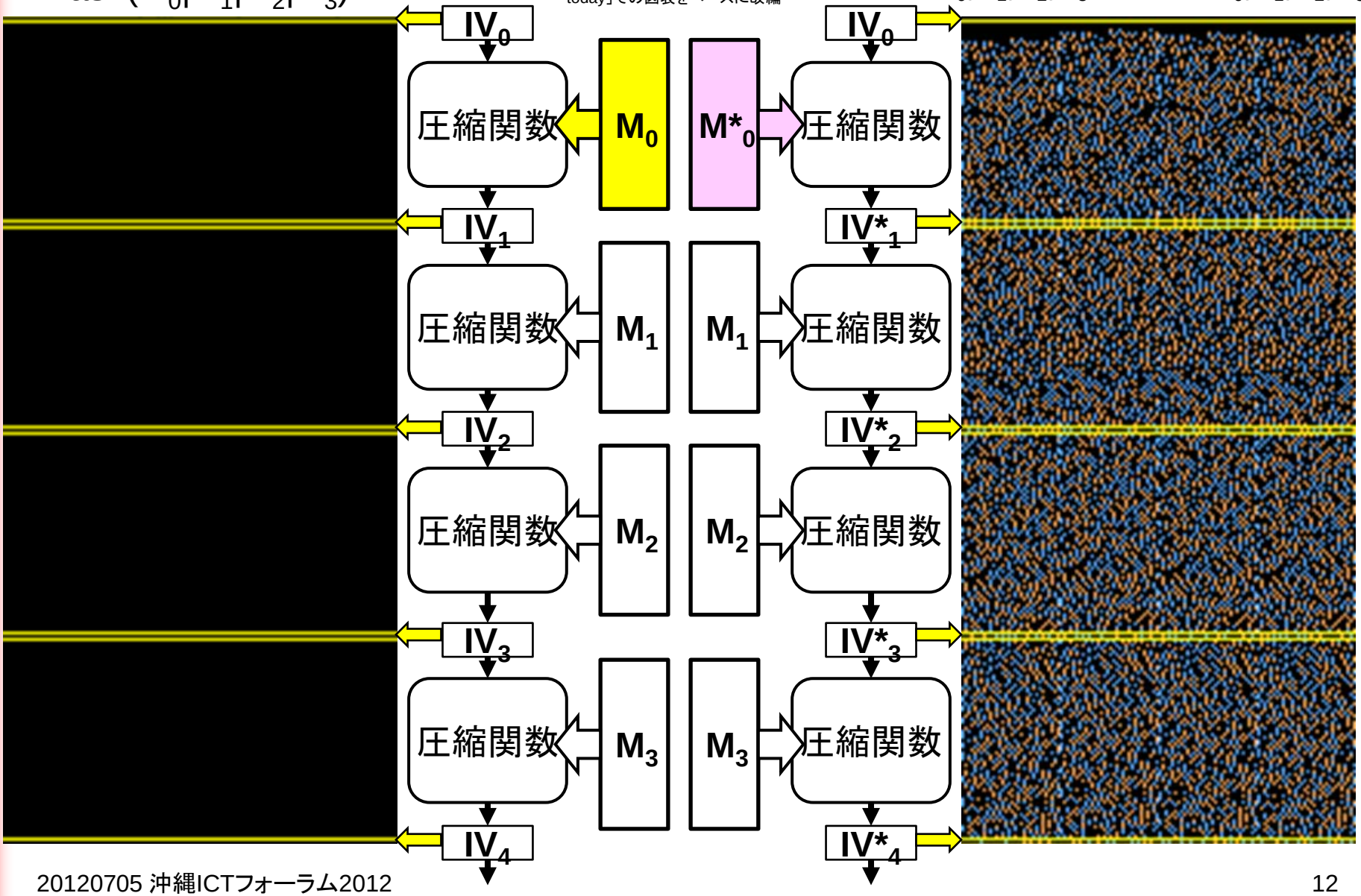


ハッシュ関数の計算の差分イメージ

Hash($M_0|M_1|M_2|M_3$)

【参考】「MD5 considered harmful today」での図表をベースに改編

Hash($M_0|M_1|M_2|M_3$)－Hash($M^*_0|M_1|M_2|M_3$)

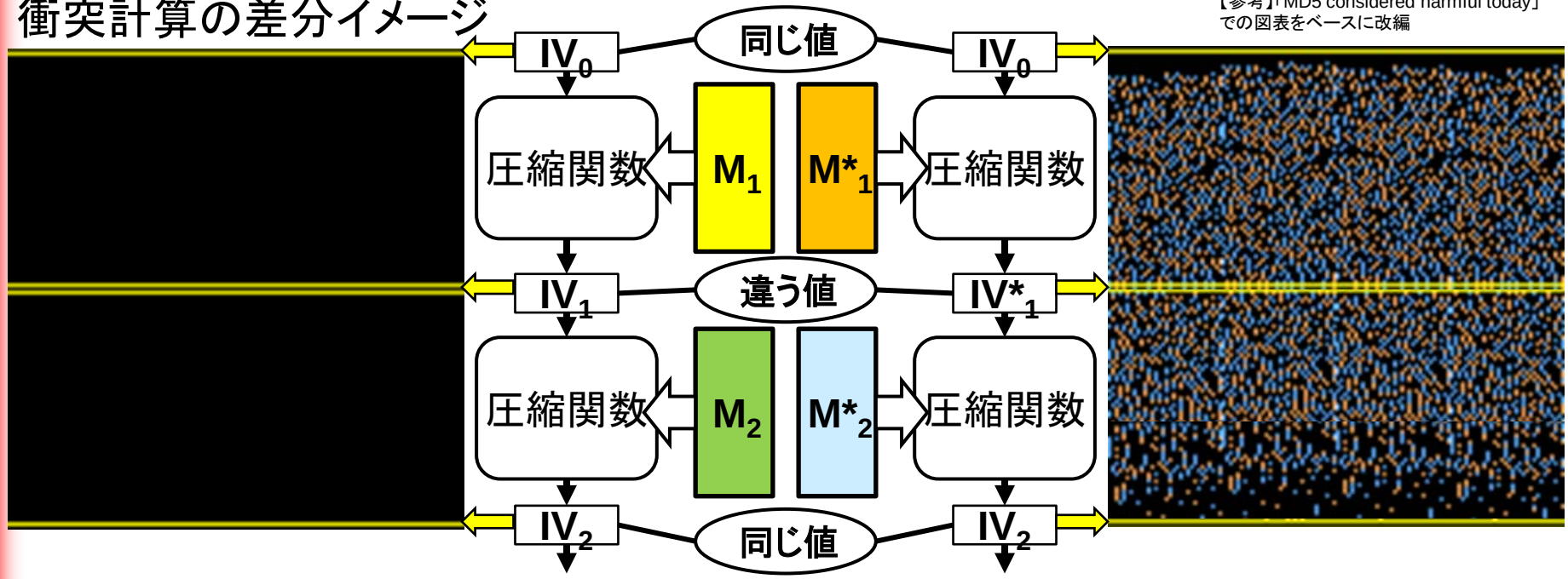


MD5（Merkle-Damgård構造）の衝突

- 2004年Wangらにより発見
  - MD5(M_1|M_2) = MD5(M^*_1|M^*_2) を見つける攻撃手法を提示
  - 計算量：約2^{39}回 → 約2^{30}回で可能（ランダムなら2^{64}回必要）

衝突計算の差分イメージ

【参考】「MD5 considered harmful today」での図表をベースに改編

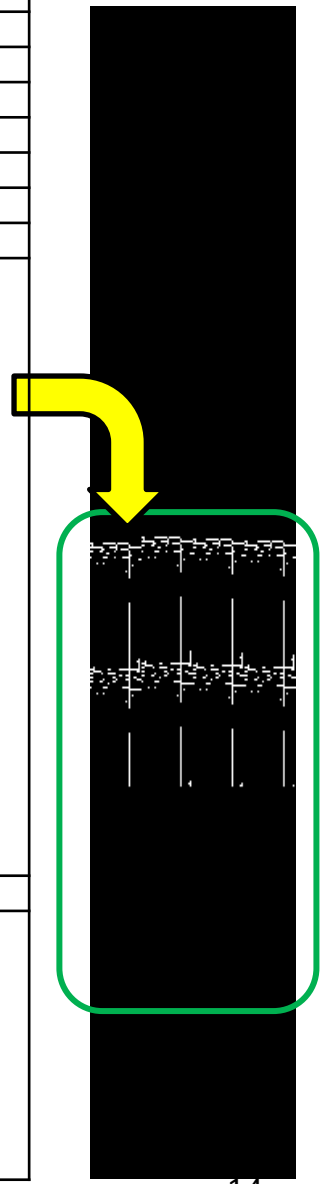


- 2005年LenstraらによりX.509偽造証明書の作成成功
  - フォーマットが正しいだけでCA署名がついているのではない

2005年のX.509偽造証明書例

	公開鍵証明書1	公開鍵証明書2
X.509 version number	0x02 (X.509 version 3)	
Serial number	0x03507449	
Signature algorithm identifier	md5withRSAEncryption	
Issuer distinguished name	CN = “Hash Collision CA”, L = “Eindhoven”, CN = “NL”	
Not valid before	Feb. 1, 2005, 00h00m01s	
Not valid after	Feb. 1, 2007, 00h00m01s	
Subject distinguished name	CN = “Hash Collision”, O = “we used a collision for MD5”, L = “Eindhoven”, C = “NL”	
Public key algorithm	rsaEncryption	
Subject public key info	3082010A 02820101 00CAB9E7 42C4B626 871AB9A5 24846B05 C18895FB **9**365E9A6 9F480392 FF2C3B3F 7941AD34 06FFADB4 034BDF84 7A4D**3**701 4FDB3283 CB19D46F A8A765C6 **B**3F016BF 306AFF7C 2E577368 9B3319B8 1564ABE7 F5B9CF66 **C**5E4FE79 0CEE047D 36CC77B0 AE5D087F 30B560EB 8872B34D 4067786**6** 2DD88464 677DBD9B 80989EF2 **4**FB82E0E A32B5864 AF33B8FE 8659B094 464699F4 77A6BFCA 348C23CF 681EC0A8 46A8B27A 29071B56 3A1316B0 5F3827B8 2FB1F9DE 1F238F3D 12AD0DDA A97DDBCF CEEAD109 395E46E0 18AE237C E59355AC 93187228 4C3A293F E9117941 A1AD5283 64A0687A FF6083B1 4B009DD9 52C866CA 43A0F41A 7DCE5876 C16CB346 E9A71809 1CEC3D57 D9020301 0001	3082010A 02820101 00CAB9E7 42C4B626 871AB9A5 24846B05 C18895FB **1**365E9A6 9F480392 FF2C3B3F 7941AD34 06FFADB4 034BDF84 7A4D**B**701 4FDB3283 CB19D46F A8A765C6 **3**3F016BF 306AFF7C 2E577368 9B3319B8 1564ABE7 F5B9CF66 **4**5E4FE79 0CEE047D 36CC77B0 AE5D087F 30B560EB 8872B34D 4067F86**5** 2DD88464 677DBD9B 80989EF2 **C**FB82E0E A32B5864 AF33B8FE 8659B094 464699F4 77A6BFCA 348C23CF 681EC0A8 46A8B27A 29071B56 3A1316B0 5F3827B8 2FB1F9DE 1F238F3D 12AD0DDA A97DDBCF CEEAD109 395E46E0 18AE237C E59355AC 93187228 4C3A293F E9117941 A1AD5283 64A0687A FF6083B1 4B009DD9 52C866CA 43A0F41A 7DCE5876 C16CB346 E9A71809 1CEC3D57 D9020301 0001
Version 3 extensions	Basic constraints	
Signature info	1319E6FF 66EF8621 AEAE0CFB D2C067B9 9C3834C0 0BE88E0A 97E60205 BC5ECD85 646B6698 BD2E9132 4826C8B1 0E2167EF F264C5E4 5A234FDE 5723A751 EA2B7913 06221B54 B4C20E4C D16562D6 98ADE4D6 33F053D6 53F8BE9C 4D402EC9 F92D3630 98DD5605 96F7BF09 5AF3C9FE D7EE2B49 21801800 3F5C65F0 511D454E 6E522913 2D0494B7 B65EF958 5AA9D433 094FDB4F 9C994610 AFE0F23F B26E5D24 6539AEFF B6E0B0DF 35B4D9AE 3CF768C5 AABC9355 8DF87BF4 21288E79 E9ADCBB8 DA236452 8E74F813 48FFB9F5 FAC43E97 4F3D79CC A222FD67 5BFD3B80 8A3F6610 4232C806 A25309A1 87D103D7 50893436 D4A32909 FE5C76B4 5495F52F 29CF66A9 E3DD473F	

差分イメージ



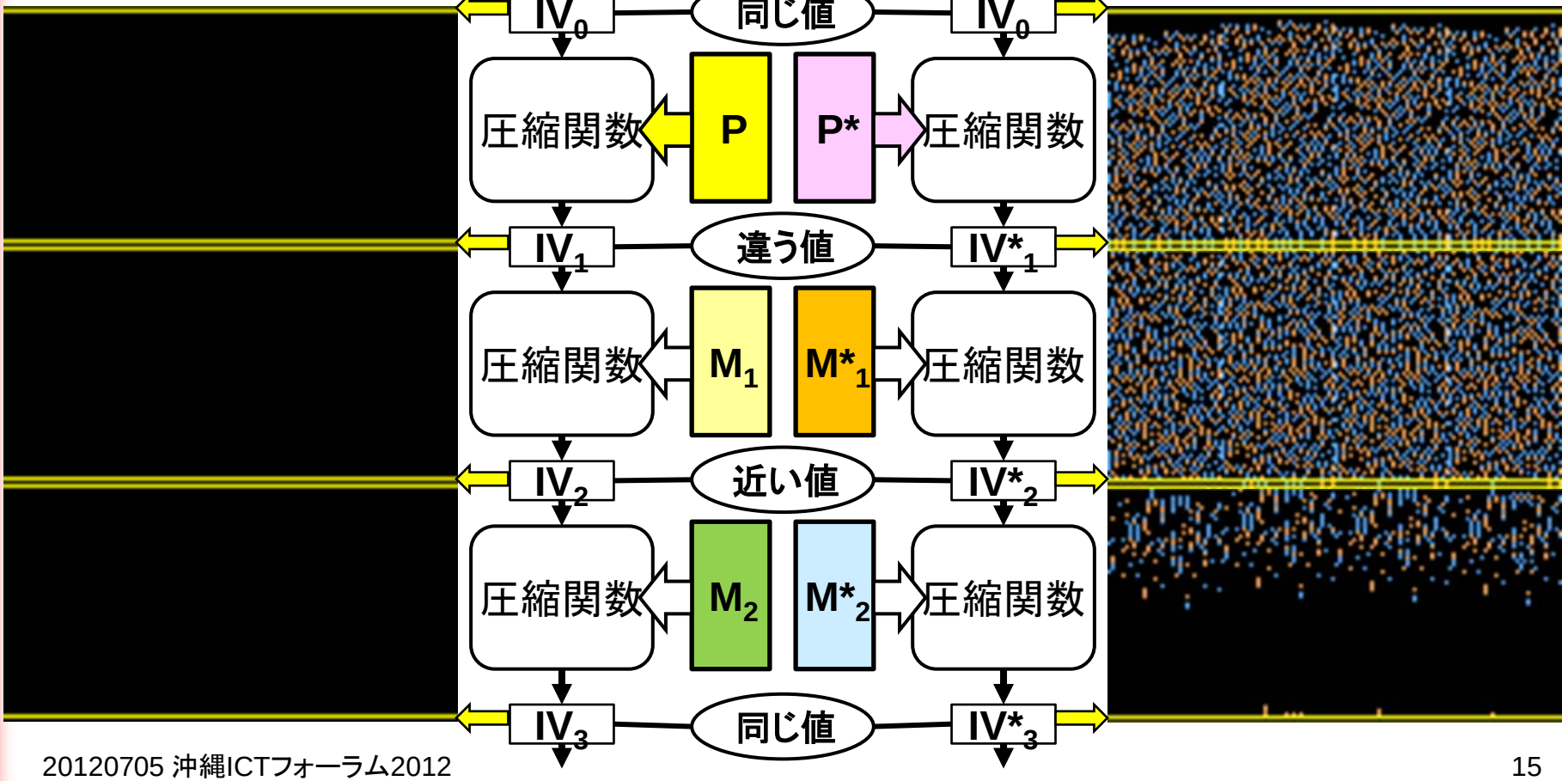
出典:「Colliding X.509 Certificates based on MD5-collisions」

かな～り精錬された攻撃に変化

- 選択プレフィックス衝突：2007年Stevensらにより発見
  - 任意の(P,P*)に対しMD5(P|M_1|M_2)=MD5(P*|M^*_1|M^*_2)となる([M_1,M_2],[M^*_1,M^*_2])を見つける攻撃手法を提示

衝突計算の差分イメージ

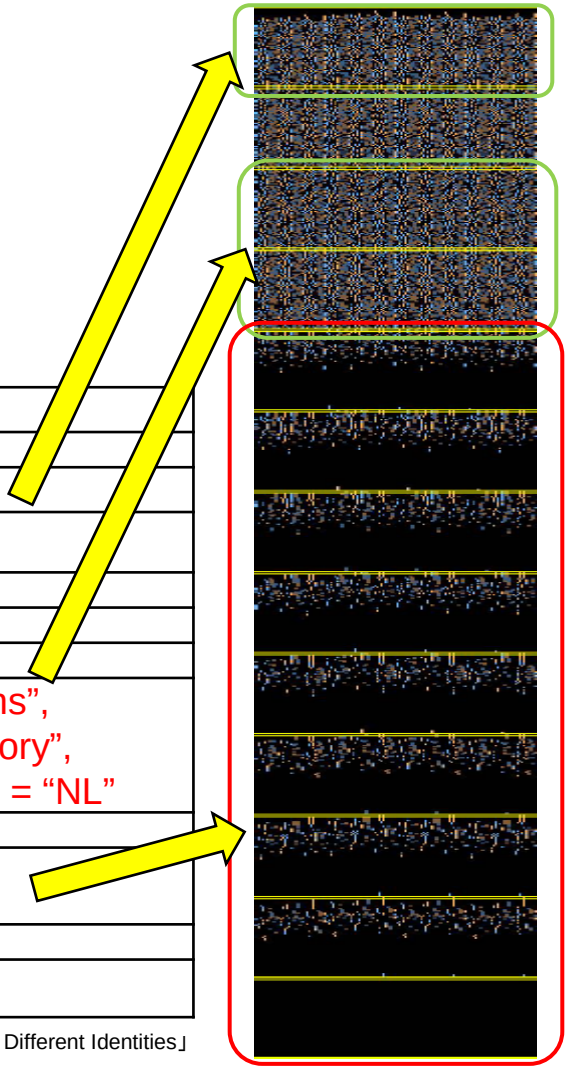
【参考】「MD5 considered harmful today」での図表をベースに改編



かな～り精錬された攻撃に変化

- 選択プレフィックス衝突
  - 制御不可のビットが存在＝衝突攻撃より制約条件が多い
  - 計算量：約2^{52}回・約6ヶ月@1200台PC
- X.509偽造証明書の例を改良
  - シリアルナンバーなどを変えた
  - CA署名がついているのではない

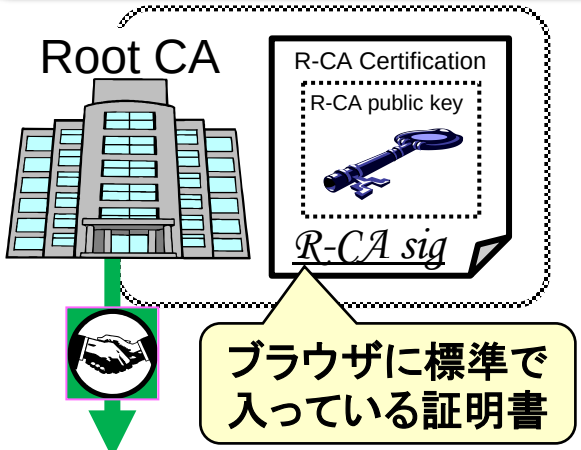
	公開鍵証明書1	公開鍵証明書2
X.509 version number	0x02 (X.509 version 3)	
Serial number	0x010C0001	0x020C0001
Signature algorithm identifier	md5withRSAEncryption	
Issuer distinguished name	CN = “Hash Collision CA”, L = “Eindhoven”, CN = “NL”	
Not valid before	Jan. 1, 2006, 00h00m01s	
Not valid after	Dec. 31, 2007, 23h59m59s	
Subject distinguished name	CN = “Arjen K. Lenstra”, O = “Collisionaris”, L = “Eindhoven”, C = “NL”	CN = “Marc Stevens”, O = “Collision Factory”, L = “Eindhoven”, C = “NL”
Public key algorithm	rsaEncryption	
Subject public key info	帳尻が合うような公開鍵情報	
Version 3 extensions	Basic constraints	
Signature info	同一の署名	



出典：「Target Collisions for MD5 and Colliding X.509 Certificates for Different Identities」

ついに本当の公開鍵証明書の偽造に成功

誘導されたら見破ることができないEE証明書が発行可能になった
⇒ MD5を利用する公開鍵証明書発行を中止する契機に



“MD5 Considered Harmful Today–Creating a rogue CA certificate”
@CCC 2008 (Chaos Communication Congress, 2008.12.30)

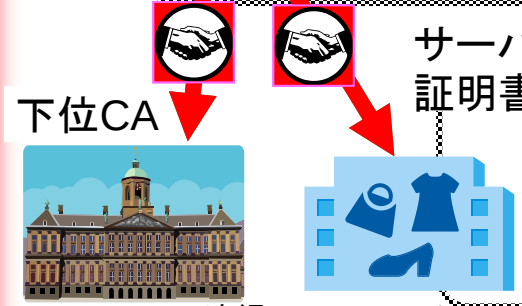
「SSL証明書の偽造」に研究者らが成功、計算には200台のPS3を使用

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]

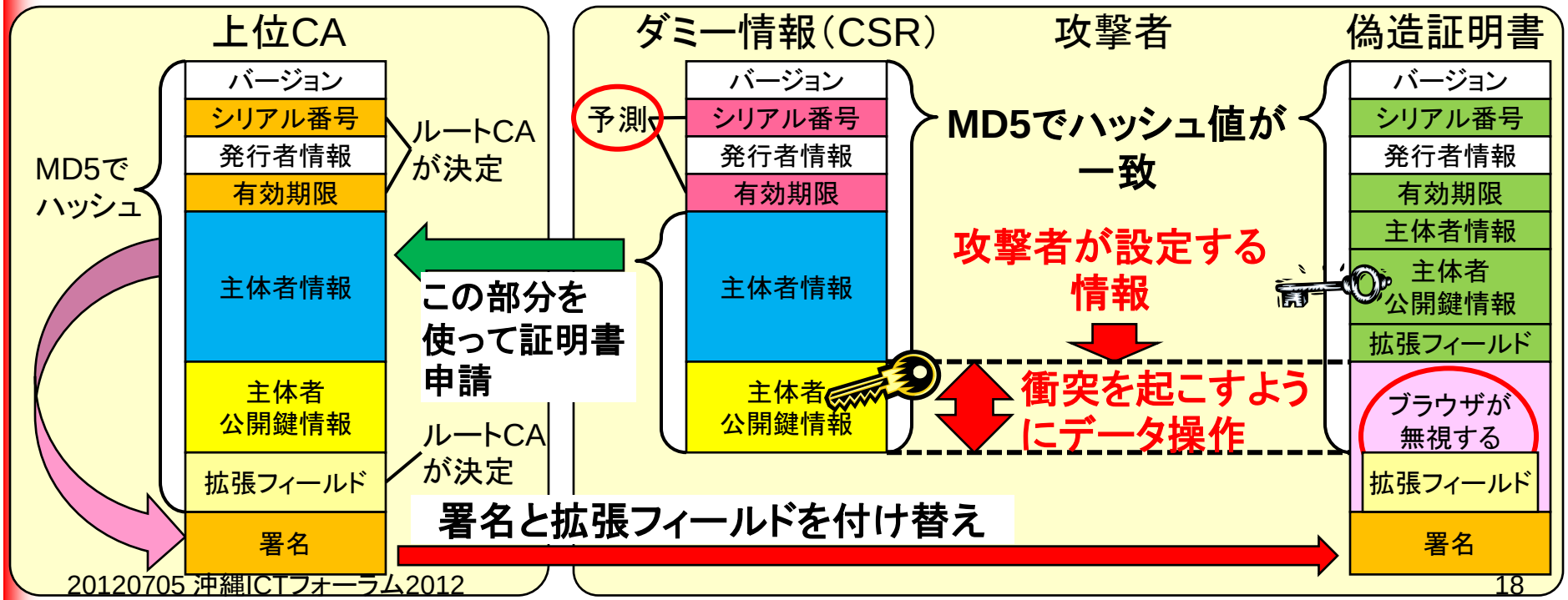
出典：日経パソコンより



中間CA 偽造EE証明書を上位認証局から発行させるのに成功

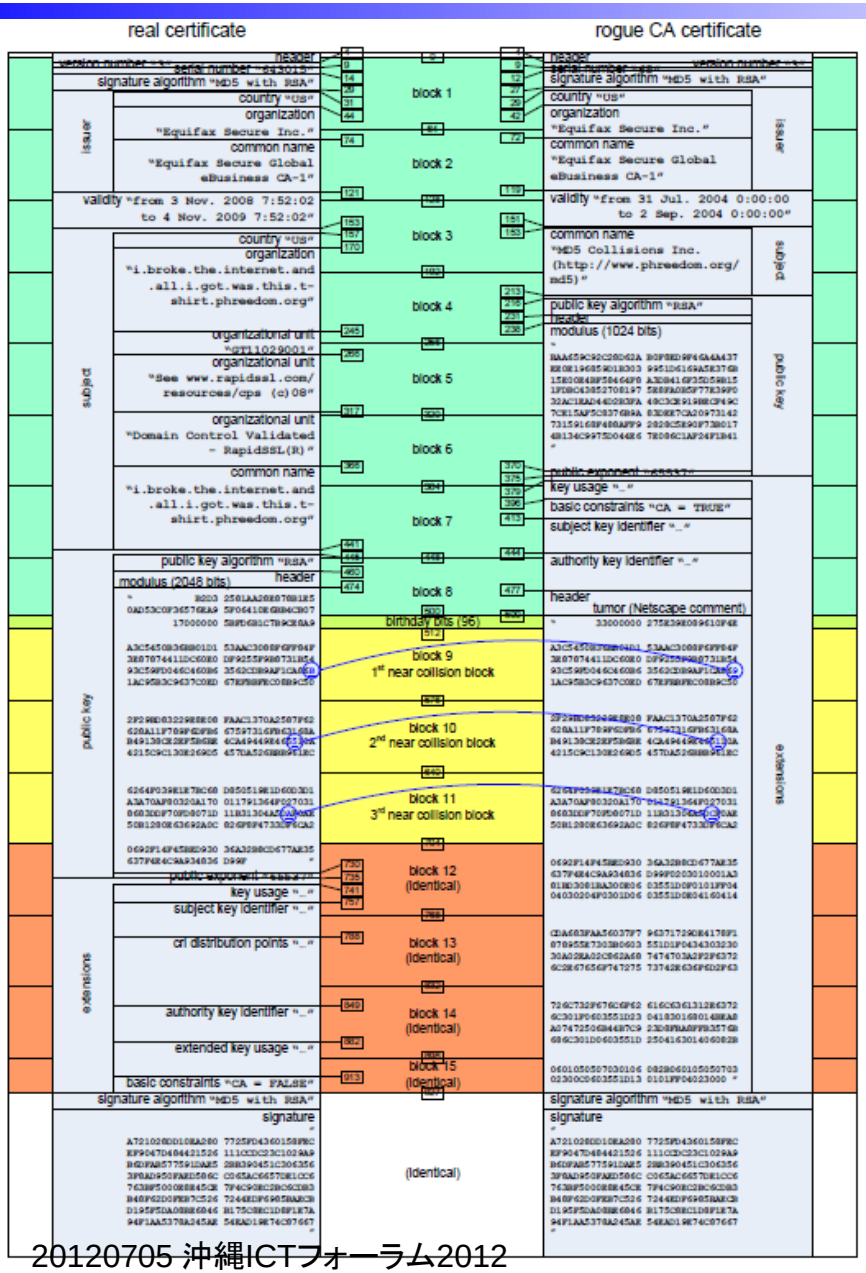
具体的にやったことは何か

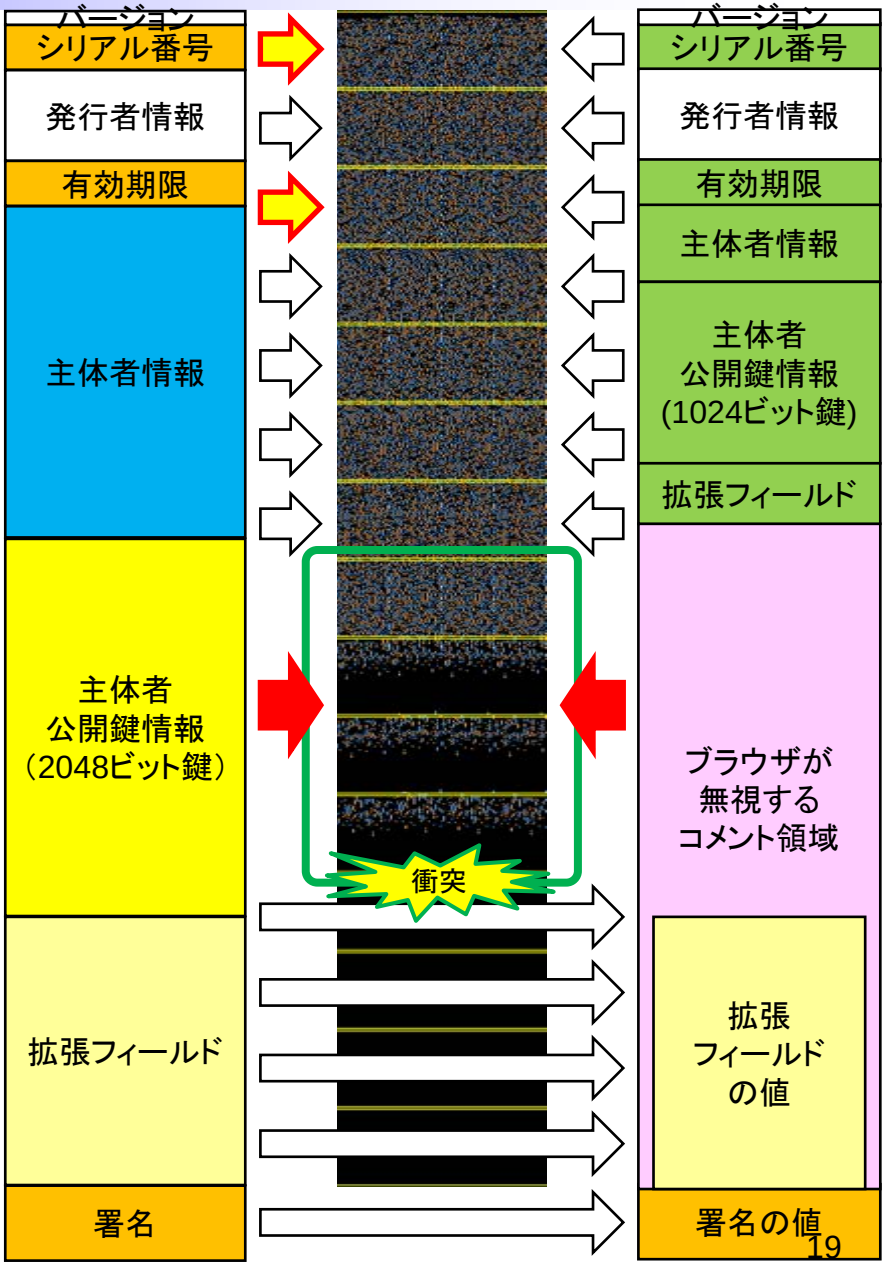
- 選択プレフィックス衝突により中間CA EE証明書偽造
  - PにRoot CAをだますためのダミー情報、P*に公開鍵を含む偽造情報を入れる
  - ルートCAが決める「シリアルナンバー」「有効期限」が予測可
  - ブラウザが無視するコメント領域を利用
  - PS3 200台で約1日



具体的にやったことは何か

出典：「MD5 considered harmful today」





さて、本題・・・

Flameで起きていたこと

一応お断り。
ここから先は色々なニュースソースをもとに神田が
総合的に理解したことをベースにしています。
誤解しているところがあるかもしれません。
そのときは遠慮なく突っ込んでください。

そもそも「Flame」って何？

- 盗聴目的に特化した完全潜伏型のマルウェア
  - 2012年5月28日　イランCERT/CCから発表
  - 約1000台感染していた
    - 少なくとも2010年2月には存在。5年前から存在していた可能性も
  - LAN/USB経由で感染・・・でも、自己増殖せず限定的な感染
    - 指示を受けての感染。自殺機能もあり

Identification of a New Targeted Cyber-Attack

Following to investigations started since 2010, about Stuxnet and Duqu, Iran National CERT MAHER) has done a technical survey during past several months. MAHER publishes) .information about the last found sample for the first time

ID: IRCNE2012051505

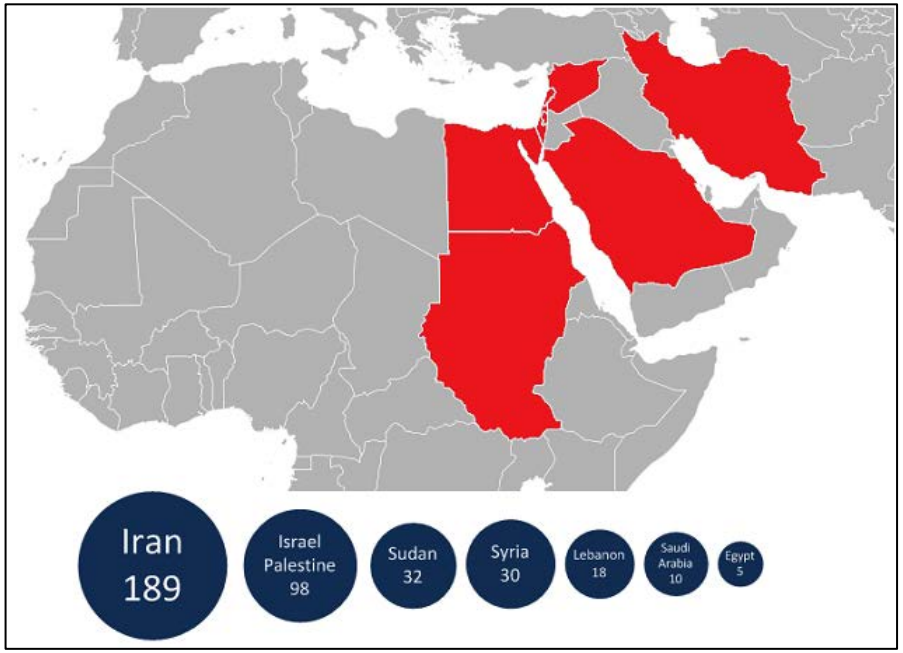
Date: 2012-05-28

Having conducted multiple investigations during the last few months, the Maher center, the Iranian CERTCC, following the continuous research on the targeted attacks of Stuxnet and Duqu since 2010, announces the latest .detection of this attack for the very first time

The attack, codenamed "Flame" is launched by a new malware. The name "Flame" comes from one of the attack modules, located at various places in the decrypted malware code. In fact this malware is a platform which is capable of receiving and installing various modules for different goals. At the time of writing, none of the 43 tested antiviruses could detect any of the malicious components. Nevertheless, a detector was created by Maher center and delivered to selected organizations .and companies in first days of May. And now a removal tool is ready to be delivered

:Some features of the malware are as follows

- Distribution via removable medias
- Distribution through local networks
- Network sniffing, detecting network resources and collecting lists of vulnerable passwords
- Scanning the disk of infected system looking for specific extensions and contents
- Creating series of user's screen captures when some specific processes or windows are active
- Using the infected system's attached microphone to record the environment sounds
- Transferring saved data to control servers
- Using more than 10 domains as C&C servers
- Establishment of secure connection with C&C servers through SSH and HTTPS protocols
- Bypassing tens of known antiviruses, anti malware and other security software
- Capable of infecting Windows Xp, Vista and 7 operating systems
- Infecting large scale local networks



出典: kaspersky Labs.
http://www.securelist.com/en/blog/208193522/The_Flame_Questions_and_Answers

そもそも「Flame」って何？

- Flameの特徴がすごい
  - マルウェアとしては非常に大きい・・・のに発見されなかった
    - 20MBもある
    - 複数のコンポーネントで構成
  - StuxnetやDuquとは兄弟的な関係
    - Stuxnetよりも20倍！も複雑 （by Kaspersky Lab.）
    - 米国とイスラエルの国家的プロジェクト（＝サイバー兵器）か？？？
  - **偽造公開鍵証明書を使ったコードサイニングがされていた**
    - **すでに知られている「MD5」に対するハッシュ衝突攻撃を利用**
    - **問題は、「MD5のハッシュ衝突探索」に対する未知の探索手法が実在し、相当ハードルが高いはずの「マイクロソフトのCA」下のPKIに入り込むことに実際に成功**
    - **偽造公開鍵証明書のトラストアンカーが「マイクロソフトのCA」**
    - **Windowsはマイクロソフトの承認を受けたコードと誤認**

      ➡ **結果として Windows Update を利用して感染？**

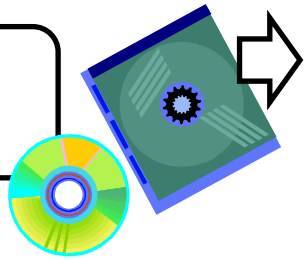
「Flame」発覚後の経緯がもっとすごい・・・

- 「Flame」が見つかった。「Stuxnet」などと比べてみよう
  - 5/28 イランCERT/CCが発表
  - 5/29 “Iran Confirms Attack by Virus That Collects Information” by New York Times
  - 5/30 “Researchers Find Clues in Malware” by New York Times
- 実は（やっぱり？）アメリカからの攻撃だったんだよね
  – Olympic Games; Bush Initiativeの暴露
  - 6/1 “Obama Order Sped Up Wave of Cyberattacks Against Iran” by New York Times
  - 6/5 “FBI Probes Leaks on Iran Cyberattack” by Wall Street Journal
- **「Flame」にMSのコードサイニングが悪用されていた**
  - **6/3 “Microsoft certification authority signing certificats added to the Untrusted Certificate Store” by Microsoft**

コードサイニング（コード署名）とは

■ ソースコードにつけるコード作成者の電子署名

ソースコード



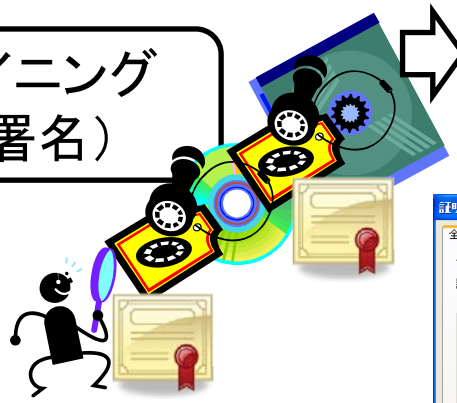
誰が本当に作ったのか確認できない。
中身が改変されていないことを確認できない。

ハッシュ関数を利用するチェックサム

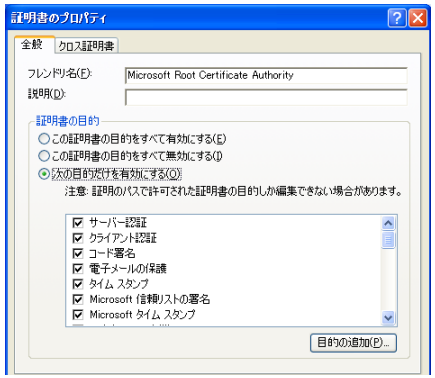
チェックサムが信用できるのであれば
（＝信用できるところにおかれているのであれば）
中身が改変されていないことを確認できる。
誰でもチェックサムは作れる。

ファイル名	作成日	リリース日	ファイルサイズ	MD5 | すべて
20120624-008-v5i32.exe | FTP	12/06/24	12/06/24	158.11 MB	9E1D99D52F14E396E822F0DC3E66B47F

コードサイニング（コード署名）

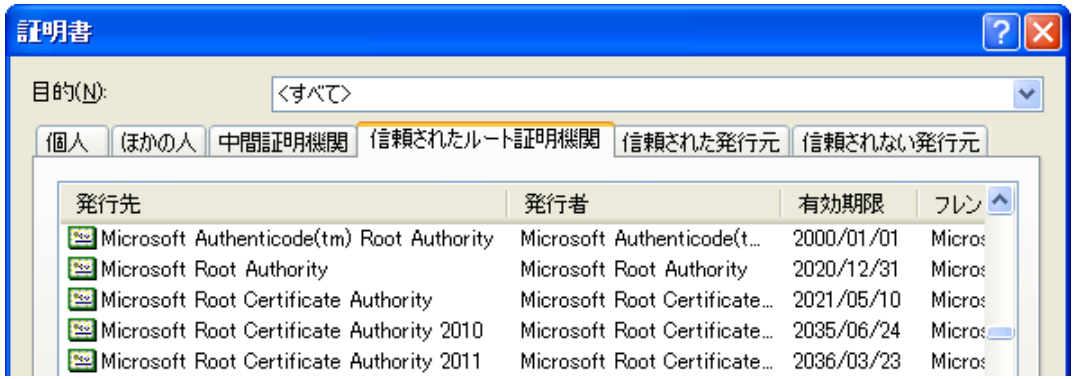


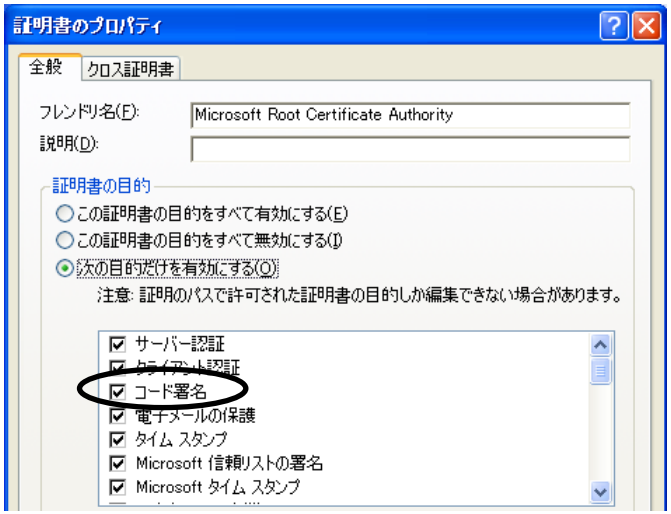
ソースコードの作成者と中身が改変されていないことを確認できる。
公開鍵証明書に対応する秘密鍵をもっている人にしか作れない。



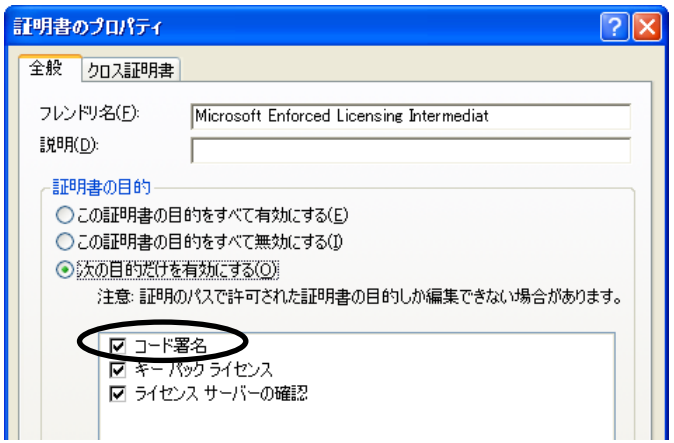
署名検証を意識しないのはなぜ？

- 実際にはOSが自動検証する
  - 「信頼されたルート証明機関」に登録された公開鍵証明書により判定

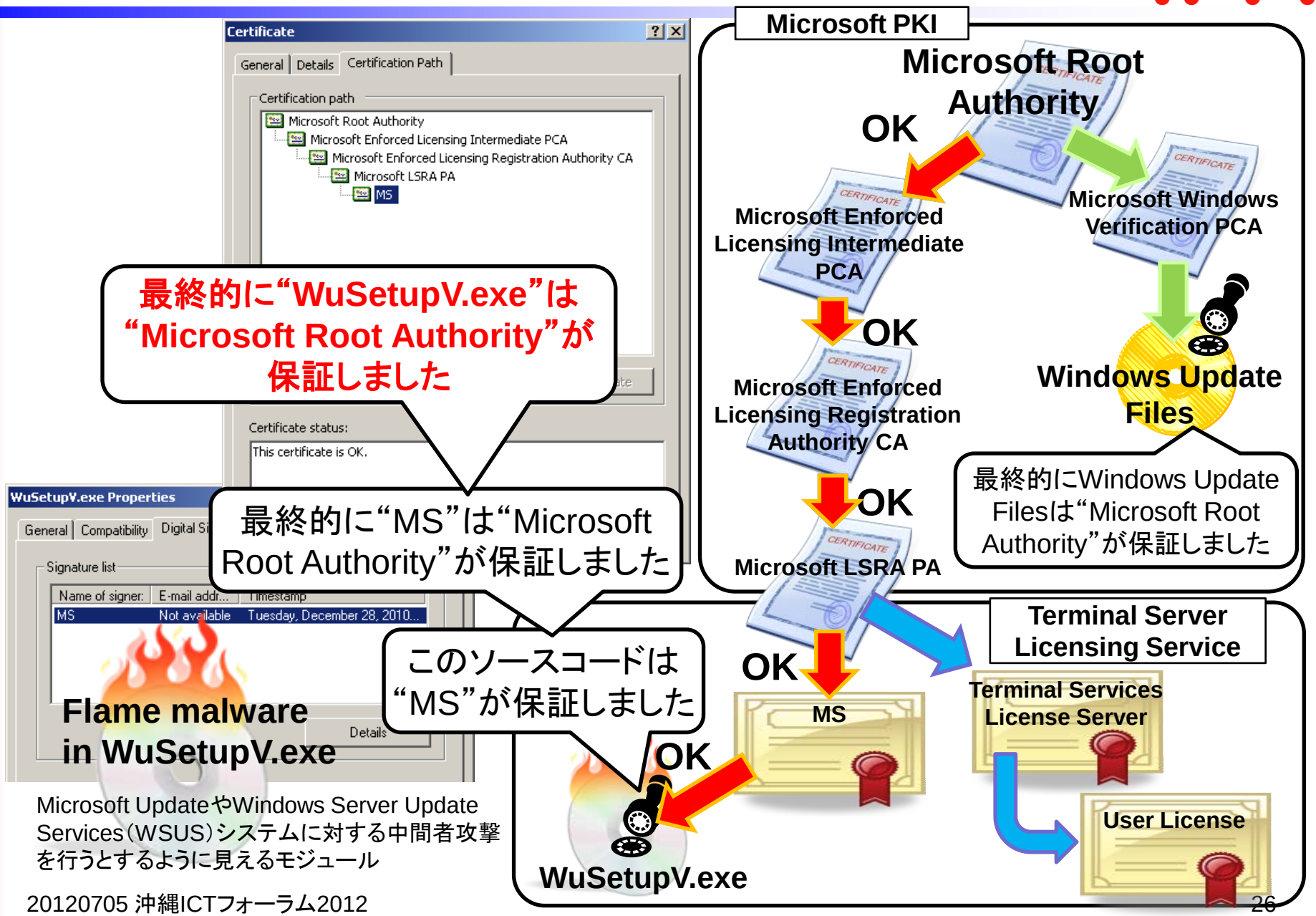




6月4日に失効させた証明書



暗号技術としてはどのように見えていたか



Microsoft UpdateやWindows Server Update Services（WSUS）システムに対する中間者攻撃を行うとするように見えるモジュール

なぜそんなことが起きたのか

- “Terminal Server Licensing Service”の本来の目的は「企業管理用の下位PKI」を作るためのもの
  - 例えばリモートログイン用クライアント証明書などの発行

- 事件発生の3大要因
  - コードサイニング（コード署名）の権限がデフォルト付与
    - Microsoft LSRA PAから公開鍵証明書を発行してもらえさえすれば、（ハッシュの衝突を利用しなくても）Windows XP以前に対する不正なコードサイニングが可能
  - “MD5”を使い続けた
    - “Microsoft Hydra X.509 extension”による防御が回避される原因
  - マイクロソフトを意味する“Microsoft Root Authority”がトラストアンカーになっていた
    - どんなソースコードでも“Windows Update filesと同等の保証”を最終的にマイクロソフトが与えてしまう仕組み

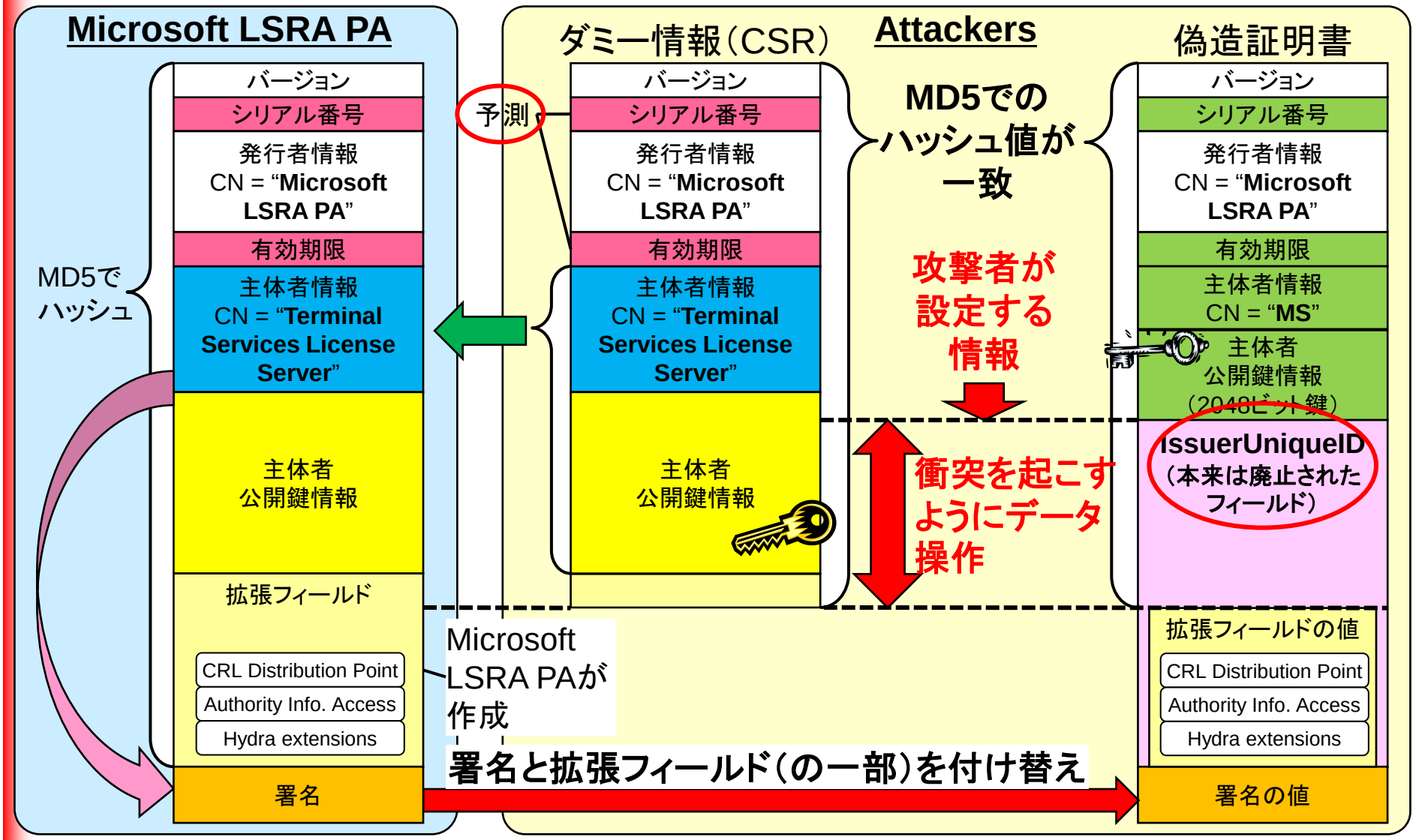
コードサイニングの必要性があったのか

マイクロソフト自身、「使うのを止めろ」と警告していたのだが・・・

PKI運用上の完全なミス

具体的に起きたこと

- 考え方は「中間CA 偽造EE証明書」の作成と同様



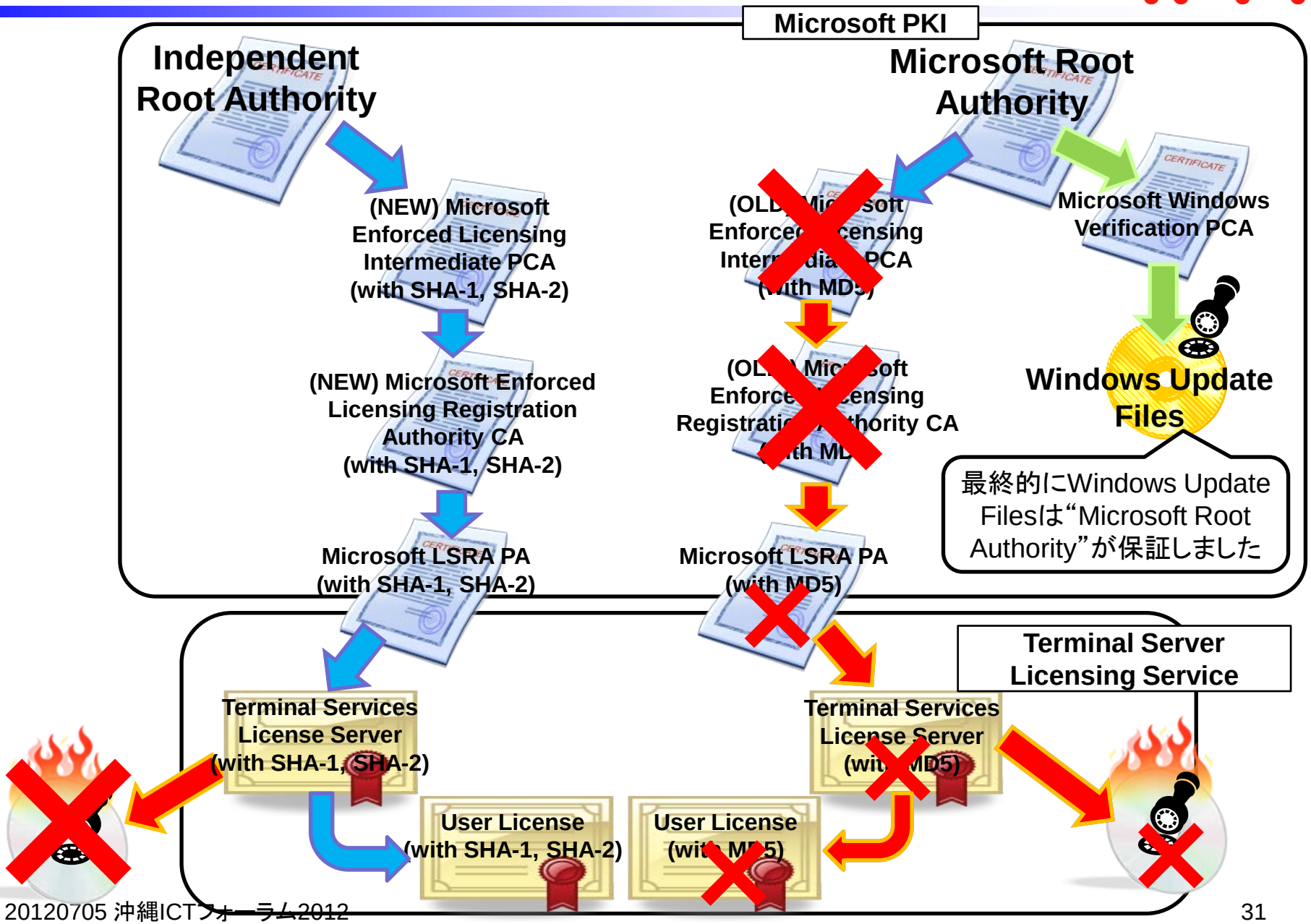
具体的に起きたこと

- issuerUniqueIDフィールドの悪用
  - Crypto APIが無視するフィールドをハッシュの衝突に利用
- MD5の悪用
  - ハッシュの衝突を利用することで、拡張フィールドが存在しない形にすることができた
- 拡張フィールドでの「Hydra extension」のチェックが機能しなかった
  - Windows XP以前のOSについてはもともと未サポート
  - Windows Vista以降のOSでは拡張フィールドが使われる際には「Hydra extensionのチェック」が必須だが、拡張フィールド自体を使うかどうかは別問題

マイクロソフトが取った対策

- セキュリティパッチの臨時配布
  - 3つの公開鍵証明書を失効（Security Advisory 2718704）
    - Microsoft Enforced Licensing Intermediate PCA (証明書 2 つ)
    - Microsoft Enforced Licensing Registration Authority CA (SHA1)
  - 新しい公開鍵証明書を配布
    - Terminal Server Licensing Serviceで使う公開鍵証明書でSHA-1, SHA-2を利用するように変更
    - ルートCAを変更（Microsoft Root Authorityではなくなった）
    - コードサイニングの禁止
  - Windows Updateエージェントの強化（KB949104, KB2720211）
    - 新しい公開鍵証明書によってコードサイニングされたファイルのみを信用するように修正

マイクロソフトが取った対策

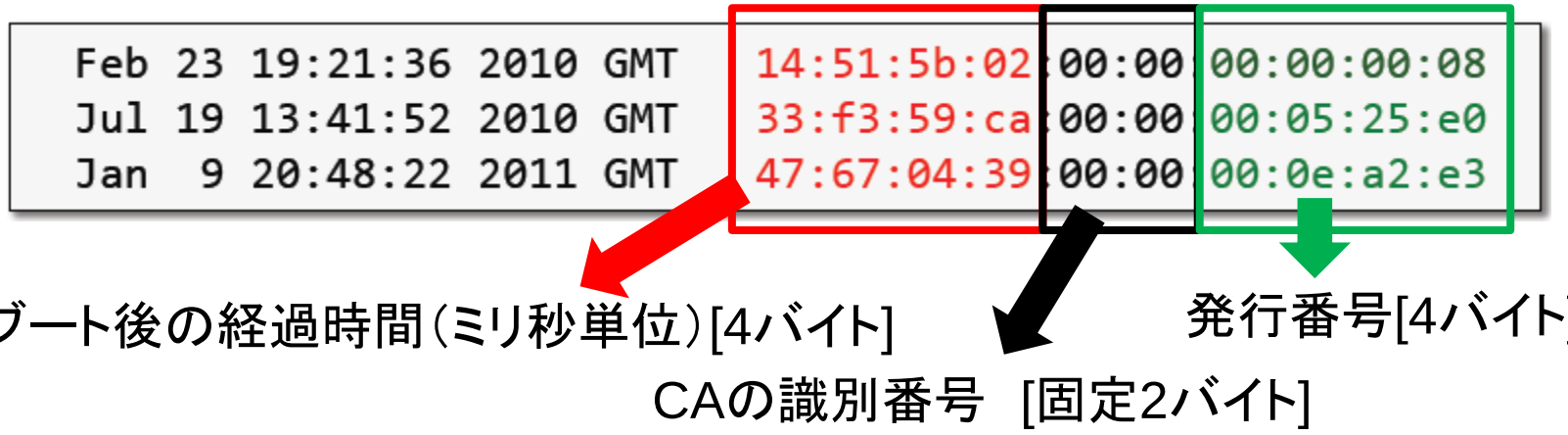


マイクロソフトが取った対策

- セキュリティパッチの定例配布
  - Windows Vista SP2, Windows Server 2008 SP2, Windows 7, Windows Sever 2008 R2用に失効した証明書を自動で処理する更新プログラムを公開（KB2677070）
    - 疑わしい公開鍵証明書を自動失効
  - （8月予定） 1024ビット未満の公開鍵証明書をブロック

もっとも“普通なら”成功可能性はかなり低い

- シリアルナンバーの予測がかなり難しい
  - Microsoft LSRA PAが作るシリアルナンバーはミリ秒単位



- 通信タイミングを合わせるのが難しい
  - ミリ秒単位で一致しないといけないので、システム負荷や通信タイミングのずれが無視できない
- 大量のハッシュの衝突データを作らないといけない
  - **現在知られている衝突探索ツールでは追いつかない**

➡ **未知の探索手法を知っている？　膨大な計算能力を持っている？**

ただ“サイバー兵器”ともなると話は別物か

- CWIによれば、少なくとも現在知られているハッシュ衝突探索手法を使ったのではない
  - 2010年には（現在でも知られていない）探索プログラムまで実際に完成していた可能性が高い
  - 同時期に異なる衝突探索手法が発見されたことを示唆
  - Microsoft LSRA PAへの不正アクセスの可能性もありうるが・・・ その場合、LSRA PAが発行する証明書フォーマットと違うことの説明がつかない
- New York Timesによれば、「Flame」は「Stuxnet – Olympic Games」とは異なる作戦だが、同時期のもの
  - 5年以上前から作戦遂行？・・・ Windows Vista以前ならここまで巧妙なことをする必要はなかったということは？？
  - サイバー兵器の乱用は避けたい・・・実は「米国がサイバー兵器に一番脆弱かもしれない」という不安

まとめに代えて

今、この瞬間にも世界のどこかでサイバー兵器が使われている・・・かも

- 「理論上起こりうる」ことは「やっぱり起きる」
  - ゼロリスクはない。国家安全保障上の問題提起と捉えるべき
    - サイバー戦争・サイバー兵器はフィクションの世界の存在ではない
    - 攻撃対象や目的は明確。目的達成のためなら“針の穴にも糸通す”

暗号アルゴリズム ＝ MD5に問題 ＋ 暗号の実装物 ＝ API処理に問題 ＋ 暗号の運用管理 ＝ PKI構造に問題

    - サイバー兵器に“ロックオン”される可能性があるなら全てを疑え
- サイバー兵器は意図なくばら撒かれるものではない
  - 「無防備に巻き込まれる」事態は避けよう
    - 直接対象となるのは例外的。ただ“予定外動作”には注意
  - （安直かもしれないが）せめてセキュリティパッチを当ててね
    - セキュリティパッチは既知のリスクを下げる役割。模倣攻撃の防止