Про інтероперабельність та безпеку Національної системи електронних цифрових підписів в Україні. По суті питання

«Але сумна історія, що її мені розказала сердешна Тарасевичівна, повинна примусити і німого говорити, і глухого слухати.
…
А я все ж волію додержувати стилю класичного;…»
Т.Г.Шевченко, «Музика (2)» (*1855. 15. I. Новопетровський форт)*

Мартиненко С.В, канд.фіз.-мат.наук

Розглянуто поточний стан питання функціональної сумісності (інтероперабельності) та безпеки Національної системи електронних цифрових підписів в Україні, проблеми та шляхи їх вирішення.

Як відомо, в електронному світі, де всі системи тісно взаємопов'язані та взаємозалежні, можливість взаємодії між програмними, програмно-апаратними продуктами різних виробників (інтероперабельність) різних систем/підсистем набуває особливого значення. Це дуже актуально для систем електронної звітності (наприклад, податкова звітність), електронного уряду, систем надання широкого спектру комерційних електронних послуг та інших електронних «масових продуктів». Завданням ефективного електронного уряду є покращення обслуговування громадян і процесів обміну інформацією (спілкування) між урядовими структурами. Для досягнення цієї мети, електронний уряд вимагає механізмів взаємодії, які дозволять ряду урядових установ пропонувати он-лайн доступ до своїх послуг і приймати участь в процедурах послуг, що надаються кількома державними установами/ відомствами (послуги «єдиного вікна»).

Питання розробки відкритих до взаємодії продуктів Національної системи електронних цифрових підписів (НСЕЦП) було і залишається гострою проблемою в Україні. Існує ряд публікацій на цю тему, де окремі проблеми висвітлено вірно. Однак, через поверхневий рівень аналізу та, можливо, нерозуміння авторами суті питання, у цих публікаціях надаються пропозиції, які здатні завести проблему в черговий тупий кут. Саме це й послугувало причиною цієї публікації.

Почнемо з узгодження деяких визначень.

Інтероперабельність, функціональна сумісність (*interoperability*, здатність до спільної роботи, *взаємодії*) в загальному розумінні – це здатність програмних, програмно-апаратних продуктів різних виробників взаємодіяти і функціонувати між собою.

Відповідно до Директиви Європейської комісії [1], *інтероперабельність* може бути визначена як «можливість обміну інформацією та взаємного використання інформації, яка мається в результаті обміну».

Більш повне технічне визначення можна знайти в документі [2] європейської програми IDABC (Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens – Інтероперабельне надання послуг європейського електронного уряду державним адміністраціям, бізнесу та громадянам): «*Інтероперабельність* означає здатність систем інформаційних і комунікаційних технологій (ІКТ) та бізнес-процесів, які вони підтримують, до обміну даними та до сумісного (колективного) використання інформації та знань». Цілями інфраструктури (Framework) європейської інтероперабельності є, зокрема, підтримання стратегії ЄС щодо забезпечення електронних послуг, орієнтованих на

користувача, шляхом взаємодії служб і систем між державними адміністраціями, а також між державними адміністраціями та громадськістю (громадянами та підприємствами) на загальноєвропейському рівні.

Слід розглядати *три аспекти* інтероперабельності [2]:

- *Організаційна інтероперабельність*. Цей аспект взаємодії стосується визначення цілей бізнесу, моделювання бізнес-процесів і забезпечення співпраці державних адміністрацій, які хочуть обмінюватися інформацією і мають (можуть мати) різні внутрішні структури і процеси. Крім того, організаційна інтероперабельність направлена на вирішення вимог співтовариства користувачів шляхом створення послуги, що відповідає вимогам, і яка є такою, що легко ідентифікується, доступна та орієнтована на користувача.
- *Семантична інтероперабельність*. Цей аспект взаємодії дозволяє гарантувати, що точне значення інформації, якою обмінялися, буде зрозумілим для будь-якої іншої прикладної програми (застосування), яка спочатку не була розроблена для цієї цілі. Семантична сумісність дозволяє системам об'єднати отриману інформацією з іншими інформаційними ресурсами і обробляти її належним чином.
- *Технічна інтероперабельність*. Цей аспект взаємодії охоплює технічні питання, пов'язані з комп'ютерними системами та послугами. Він включає в себе основні аспекти, такі як відкриті інтерфейси, служби взаємодії (interconnection services), інтеграція даних і допоміжне програмне забезпечення (middleware), представлення даних (data presentation) і обмін даними, доступність і безпека послуг.

Таким чином, впровадження послуг електронного уряду на європейському рівні вимагає розгляду проблем взаємодії з точки зору *організаційних, семантичних і технічних аспектів*.

На додаток до загальноєвропейських стандартів та програм слід зазначити, що існують і відповідні національні програми/ стандарти, наприклад, у Німеччині SAGA (Standards and Architectures for e-government Applications – Стандарти та архітектури для застосувань електронного уряду) [3] та інші.

Не розглядаючи в межах цієї публікації два перших із зазначених аспектів, які є принципово важливими, зробимо лише коротке зауваження. На сучасному етапі окремі рекомендації щодо *інтероперабельності в* НСЕЦП України зводяться до впровадження деякого «універсального транспортного/ цифрового конверту», як панацеї для вирішення усіх проблем. При цьому «транспортний конверт» (наприклад, як схема XML документу) створюється/ розробляється заново (для використання виключно в Україні), не аналізуючи та не використовуючи вже існуючі міжнародні/ європейські стандарти щодо таких систем обміну, і подається без будь-якого моделювання бізнес-процесу, без створення основного та альтернативного потоку документів (у термінах мови UML, див., наприклад, стандарти S.W.I.F.T., ISO 20022, специфікації програми IDABC) тощо. В результаті виявляється відсутність принципово важливих компонентів/ складових процесів, наприклад, процедури відкликання/ скасування помилково відправленого документу, не регламентовано дії у позаштатних ситуаціях та при виникненні помилок на будь якому етапі життєвого циклу тощо.

Сучасною «модою» стало використання в якості «транспортних конвертів» XML документів. Але при цьому, як правило не надаються XML схеми, не визначається простір імен, не використовуються стандарти (формати) цифрового підпису та шифрування XML документів тощо. Прикладами «транспортних конвертів» українського виробництва є, зокрема, відповідні конверти податкової звітності ДПА України (наказ від 11.02.2011 р. № 90), та взагалі органів державної влади (Наказ МОНмолодьспорт від 20.10.11 р. № 1207) та інш. Але, якщо у специфікації XML документу відсутні XSD схема, та/чи такі атрибути як «простір імен» (namespace, targetNamespace), «кодування» (encoding) та деякі інші, якщо у специфікації

присутні «унікальні» ідентифікатори (ID) без чіткого визначення правил формування їх значень (де їх брати, хто їх призначає …), то для інтероперабельних систем цю специфікацію не варто навіть читати, не кажучи вже про її реалізацію. Наприклад, «простір імен» (namespace) призначається для локалізації імен атрибутів в межах схеми документу. Для інтероперабельності ми маємо на меті застосування розширюваної мови розмітки (XML), де один XML документ може містити елементи й атрибути (називаються «словник розмітки»), які визначені для використання кількома модулями програмного забезпечення. Однією з вимог модульності при використанні словників розмітки є відсутність проблем «розпізнання» (recognize) та «зіткнення/ колізій» (collision). Тобто, програмні модулі повинні бути здатні розпізнавати теги та атрибути, які необхідно обробляти, навіть в умовах «зіткнення», що виникають при розмітці, призначеній для іншого пакету програмного забезпечення, який використовує той же тип елемента або те ж саме ім'я атрибута. Для унеможливлення проблем розпізнання та колізій саме й призначається «простір імен» (namespace).

Висновок можна зробити одразу – впровадження такого «транспортного конверту» не тільки не наблизить, а навпаки, зашкодить інтероперабельності в НСЕЦП України. Не слід винаходити українські «універсальні транспортні конверти» та протоколи, потрібно використати відповідні міжнародні/ європейські стандарти (див. далі), які вже пройшли як теоретичний аналіз, так і практичну апробацію.

На завершення *організаційної та семантичної інтероперабельності* слід також підкреслити, що і Національний банк України не може бути осторонь цього процесу. Так як ряд послуг електронного уряду може бути (чи вже є) платними, а також враховуючи те, що застосування фінансових інструментів вимагають такі системи, як електронна комерція, електронна митниця, контроль за поверненням ПДВ тощо, НБУ також необхідно розробити відповідні технічні специфікації (перелік стандартів) та вимоги щодо інтероперабельності. В основі специфікацій повинні бути міжнародні/ європейські стандарти, наприклад, відповідні специфікації програми IDABC і, звичайно, ISO 20022.

Для подальшого розгляду проблем *Технічної інтероперабельності,* будемо вважати, що два перші із зазначених вище аспектів інтероперабельності відповідним чином (професійно) вже розглянуто та вирішено. Отже, як зазначено вище, *Технічна інтероперабельність* включає в себе [2]:

- відкриті інтерфейси,
- служби міжсистемного зв’язку/ взаємодії (interconnection services),
- інтеграцію даних і допоміжне програмне забезпечення (middleware),
- представлення даних (data presentation) і обмін даними,
- доступність,
- безпеку послуг у цілому, та конфіденційність персональних даних зокрема.

Основними *принципами інтероперабельності* є (класично):

- Відкритість інтерфейсів,
- Слідування стандартам,
- Транспортабельність даних.

Із складових технічної інтероперабельності тут розглянемо детально лише такі ключові складові НСЭЦП, як:

а) *криптографічні модулі* – програмні, програмно-апаратні засоби криптографічного захисту інформації (КЗІ);

б) *формати об’єктів системи цифрових підписів і шифрування* – формати X.509 сертифікатів, цифрового підпису/ шифрування та інших криптографічних об’єктів НСЭЦП.

Відкритість інтерфейсів (перший принцип інтероперабельності) означає, що засіб КЗІ має відкритий «Інтерфейс прикладного програмування» (Application Programming Interface, API) – набір класів, процедур, функцій, структур і констант, що надаються засобом КЗІ для використання у зовнішніх програмних продуктах. API використовується для написання

прикладних програм (застосувань). Слід зазначити, що термін «відкритість» API означає, що інтерфейс засобу КЗІ відповідає певним стандартам (принцип слідування стандартам) для засобів КЗІ.

Дотримання стандартів (другий принцип інтероперабельності) – це, окрім стандартів API, також, і навіть головним чином, слідування стандартам безпеки для криптографічних модулів. Такими міжнародно відомими та визнаними стандартами є стандарт США «FIPS 140-2» [4] і міжнародні стандарти серії «Загальні критерії оцінки захищеності інформаційних технологій (Common Criteria for Information Technology Security Evaluation), або більш коротка назва «Загальні критерії» (Common Criteria, CC) – ISO/IEC 15408-1: 2009 [5], ISO/IEC 15408-2:2008 [6], ISO/IEC 15408-3:2008 [7]. В межах Європейського Союзу застосовуються додатково стандарти ДСТУ CWA 14365-2:2009 [8], ДСТУ CWA 14167-3: 2008 [9], ДСТУ-П CWA 14172-5:2008 [10], ДСТУ-П CWA 14172-7:2008 [11], ДСТУ CWA 14355:2009 [12], CWA 14169:2004 [13], CWA 14170:2004 [14].

Слід зазначити, що інтероперабельність і безпека по суті є антагоністичними поняттями – чим вище вимоги безпеки, тим, як правило, значно важче забезпечити інтероперабельність, і навпаки. Для засобів КЗІ потрібно об'єднати інтероперабельність і безпеку найбільш оптимальним чином, тільки у цьому випадку отримаємо юридично значущий електронний документообіг.

Транспортабельність даних (третій принцип інтероперабельност) щодо засобів КЗІ означає, зокрема, що при заміні одного засобу КЗІ на інший забезпечується можливість експорту/імпорту даних, що зберігаються в них, тобто криптографічних ключів, X.509 сертифікатів і т.інш.

Інтероперабельність НСЕЦП, у розрізі складових, які тут розглядаються, по суті означає, що:

1) документи (дані), підписані цифровим підписом і зашифровані із застосуванням криптографічного модуля одного виробника, можуть бути розшифровані і перевірено підпис із застосуванням криптографічного модуля іншого виробника; аналогічно й інші криптографічні об'єкти, такі як X.509 сертифікати, списки відкликання (CRL), позначки часу і т.інш.;

2) прикладна система (застосування), розроблена для роботи з криптографічним модулем одного виробника, не потребує доопрацювань (модернізації, внесення змін і т.інш.) при переході до/додаванні криптографічного модуля іншого виробника.

Розглянемо ключові проблеми (бар'єри), поточний стан інтероперабельності та безпеки НСЕЦП Україні, окремі публікації на цю тему і надані у них пропозиції (рекомендації).

Проблема №1. Невідповідність законодавства України законодавству ЄС

Базовим європейським законодавчим актом є Директива 1999/93/ЄС [16], а в Україні відповідно Закон України «Про електронний цифровий підпис» [17].

Наведемо порівняльну таблицю основних розходжень зазначених законодавчих актів у базових термінах:

Закон України	Директива ЄС (офіційний переклад)
електронний підпис – дані в електронній формі, які додаються до інших електронних даних або логічно з ними пов'язані та призначені для ідентифікації підписувача цих даних;	*електронний підпис* – дані, поданні в електронній формі, які додаються або логічно об'єднуються з іншими електронними даними та які служать в якості метода засвідчення достовірності. *Коментар. Це офіційний переклад, але «метод засвідчення достовірності» в*

Закон України	Директива ЄС (офіційний переклад)
	оригіналі – це «method of authentication», що в інформаційних технологіях означає: *- перевірити цілісність даних, та* *- ідентифікувати підписувача даних.* *Див., наприклад, Barron's Banking Dictionary: «Authentication – Legal verification of the genuineness of a bond, document, or signature. In electronic funds transfers, authentication is a method of verifying that a payment instruction has in fact originated at the sending bank, and has not been tampered with by an unauthorized party» (Правова перевірка справжності зобов'язання, документа або підпису. В електронному переказі коштів, аутентифікація - це метод перевірки того, що платіжне доручення, по суті, виникло в банку-відправнику, і не було змінено сторонніми особами).*
електронний цифровий підпис – вид електронного підпису, отриманого за результатом криптографічного перетворення набору електронних даних, який додається до цього набору або логічно з ним поєднується і дає змогу підтвердити його *цілісність* та *ідентифікувати* підписувача.	Термін відсутній *Коментар. За змістом ЕЦП – це дещо обмежений електронний підпис Директиви ЄС. Обмежується тим, що він отримується за результатом криптографічного перетворення.*
Термін відсутній *Коментар. Сумісність ЕЦП з визначенням «удосконалений цифровий підпис» тільки за пп. (b) та (d)*	***удосконалений електронний підпис*** означає електронний підпис, який відповідає наступним вимогам: (a) він пов'язаний винятково з особою, що підписалась; (b) він дає можливість ідентифікувати особу, що підписалась; (c) він створений за допомогою засобів, які особа, що підписалась, може тримати під своїм повним контролем; і (d) він пов'язаний з даними, до яких він відноситься у такий спосіб, що будь-яку подальшу зміну даних можна виявити;
надійний засіб електронного цифрового підпису – засіб електронного цифрового підпису, що має сертифікат відповідності або позитивний експертний висновок за результатами державної експертизи у сфері криптографічного захисту інформації. *Коментар: Визначення, наведене в Законі України, не вимагає виконання вимог (a), (c)*	*безпечний механізм створення підпису* - означає механізм створення підпису, який відповідає вимогам, викладеним у Додатку III: a) дані, які використовуються для вироблення підпису, можуть виникнути на практиці лише один раз, а їх секретність забезпечується; b) дані, які використовуються для

Закон України	Директива ЄС (офіційний переклад)
Директиви ЄС для удосконаленого електронного підпису. Визначення Закону дозволяє довільну трактовку «надійності» (у тому числі, на рівні підзаконних актів).	вироблення підпису, із значною долею впевненості не можуть вилучатися з цих механізмів, а підпис захищається від підробки за допомогою використання доступних технологій; c) дані, що створюють підпис, які використовуються для вироблення підпису, можуть бути надійно захищені законною особою, що підписалась від використання його іншими особами.
посилений сертифікат відкритого ключа (далі – посилений сертифікат ключа) – сертифікат ключа, який відповідає вимогам цього Закону, виданий акредитованим центром сертифікації ключів, засвідчувальним центром, центральним засвідчувальним органом.	Термін відсутній
Термін відсутній	термін «*кваліфікований сертифікат*» (qualified certificate) означає сертифікат, який відповідає вимогам, викладеним у Додатку I (Вимоги до кваліфікованих сертифікатів), і видається постачальником послуг сертифікації, який виконує вимоги, викладені у Додатку II (Вимоги до постачальників послуг сертифікації, що видають кваліфіковані сертифікати);
Електронний цифровий підпис за правовим статусом прирівнюється до власноручного підпису (печатки) у разі, якщо: електронний цифровий підпис *підтверджено з використанням посиленого сертифіката ключа за допомогою надійних засобів цифрового підпису;*	Держави-члени забезпечують, щоб *удосконалені електронні підписи, засновані на кваліфікованих сертифікатах і створені за допомогою безпечних механізмів створення підпису*: a) задовольняли юридичним вимогам до підписів стосовно даних, поданих у електронній формі, так само, як підпис, написаний власноручно, задовольняє вимоги стосовно даних, нанесених на папір; і b) були прийнятними в якості доказів у судочинстві.

Отже, маємо таку невідповідність юридичної сили/ правового статусу:

Закон України	Директива ЄС
1. електронний цифровий підпис *підтверджено* з використанням 2. *посиленого* сертифіката ключа 3. за допомогою *надійних* засобів цифрового підпису;	1. *удосконалені* електронні підписи, 2. засновані на *кваліфікованих* сертифікатах і 3. *створені* за допомогою 4. *безпечних* механізмів створення підпису

Таким чином, маємо такі основні невідповідності:

1) «посилений сертифікат» Закону України не відповідає «кваліфікованому сертифікату» Директиви ЄС,

2) «надійний засіб ЕЦП» Закону України не відповідає «безпечному механізму створення підпису» Директиви ЄС.

3) За Законом України для набуття юридичної сили підлягає контролю (та вимагається безпека) лише етап перевірки підпису («електронний цифровий підпис підтверджено з використанням …», а Директива вимагає контроль на етапі створення («створені за допомогою …»).

Отже, базові терміни Закону України не відповідають термінам Директиви ЄС, немає спільних/ сумісних критеріїв та відповідності щодо юридичної сили/ правового статусу електронного (цифрового) підпису між Законом України і Директивою ЄС.

Найбільш критичним є значно знижений (законодавчо) рівень безпеки ЕЦП України у порівняння з цифровим підписом ЄС, що віддзеркалюється на інших законодавчих та нормативних актах.

До речі, у Росії свого часу було прийнято аналогічний закон (від 08.11.2007 р., попередній у 2002), який мав такі ж недоліки неврегульованості визначень та термінів між законодавством РФ та ЄС. Ці недоліки було враховано та усунено в новій редакції Закону «Про електронний підпис» від 2010 року, де визначено три види електронного підпису - *простий, посилений та кваліфікований.*

Проблема №2. Низький рівень стандартизації НСЕЦП. Тестування відповідності у сфері КЗІ

Про експертизу криптографічних засобів

Відсутні чіткі технічні вимоги (програми тестування, тестових векторів тощо), яким повинні задовольняти криптографічні засоби з метою отримання сертифікату відповідності або позитивного експертного висновку за результатами державної експертизи у сфері криптографічного захисту інформації (КЗІ). На поточний момент експертиза здійснюється на відповідність технічному завданню виробника, яке попередньо підлягає погодженню з контролюючим органом у сфері КЗІ. Технічне завдання – це «суб'єктивний» документ (для одного й того ж засобу КЗІ може бути викладено на 20 чи на 200 аркушах, відповідно до вимог міжнародних стандартів чи «на власний розсуд/ розуміння»), а тому й позитивний експертний висновок є більше «суб'єктивним» документом, який не відображає дійсного рівня відповідності, надійності та безпеки засобу КЗІ.

Кожен ліцензіат складає та затверджує в контролюючому органі у сфері КЗІ власну методику перевірки правильності реалізації (тестування). Такі методики є також «суб'єктивними», бо складаються «на власний розсуд/ розуміння» ліцензіата. Повнота та комплексність цих методик, як правило, знаходиться на незадовільному рівні. Наприклад, найбільш поширена та найбільш, як вважається, детальна методика тестування правильності реалізації ДСТУ 4145 містить 266 тестових вектори. Але у тестах «приймають участь» лише три еліптичні криві поліноміального базису (173, 283, 431 біт) із 10-ти визначених стандартом та три еліптичні криві оптимального нормального базису (173, 233, 431 біт) із 5-ти визначених стандартом. Таким чином, будь яка помилка у реалізації (даних) інших, не охоплених тестами та не перевірених, еліптичних кривих не може бути виявлена на етапі тестування. У результаті, якщо засіб і має позитивний експертний висновок, але правильність його реалізації, а отже і інтероперабельність, є сумнівною.

Хорошою державною практикою є створення відповідними державними органами та відкрите опублікування тестових векторів і методик тестування. Так, наприклад, Національний Інститут Стандартів і Технології США (NIST, National Institute of Standards and Technology) на потреби промисловості та користувачів в об'єктивних, незалежних тестах для інформаційних технологій, що призвані допомагати компаніям випускати наступне покоління продуктів і послуг, розробляє серії криптографічних тестів та тестів інтероперабельності PKI (Public Key Interoperability), зокрема, в межах таких програм:

- NIST програма валідації/ перевірки правильності криптографічних алгоритмів CAVP (Cryptographic Algorithm Validation Program) та
- NIST програма набору тестів інтероперабельності відкритих ключів PKITS (Public Key Interoperability Test Suite).

Розроблені тести постійно переглядаються та оновлюються з урахування набутого досвіду при виявленні помилок та невідповідностей.

Як приклад, можна звернутися до публікацій [20, 21] та інш., а також до відповідних серій тестових векторів, зокрема:

- Алгоритму шифрування AES:
  - AES Known Answer Test (KAT) Vectors
  - AES Monte Carlo Test (MCT) Sample Vectors
  - AES Monte Carlo Test (MCT) Intermediate Values
  - AES Multiblock Message Test (MMT) Sample Vectors
- Алгоритму шифрування TDES:
  - Triple DES Known Answer Test (KAT) Vectors
  - Triple-DES Monte Carlo Test (MCT) Sample Vectors
  - Triple-DES Monte Carlo Test (MCT) Intermediate Values
  - Triple-DES Multiblock Message Test (MMT) Sample Vectors

Аналогічно, існують тестові вектори для асиметричних алгоритмів RSA (ANSI X9.31), DSA (Digital Signature Algorithm), ECDSA (Elliptic Curve DSA; ANSI X9.62):

- 186-2 DSA Test Vectors
- 186-2 RSA Test Vectors
  - RSA SigVer PKCS1.5 Vulnerability Test Vectors
  - RSA SigVer X9.31 Vulnerability Test Vectors
- 186-2 ECDSA Test Vectors
- 186-3 DSA Test Vectors
- 186-3 RSA Test Vectors
- 186-3 ECDSA Test Vectors.

та генераторів випадкових чисел, інших криптографічних алгоритмів та протоколів.

Призначення цих тестів полягає у наданні допомоги компаніям виробляти інтероперабельні сумісні компоненти систем Інфраструктури відкритих ключів (PKI), тобто криптографічних засобів. Створені набори тестів дозволять розробникам та тестовим лабораторіям визначити відповідність програм/ продуктів PKI стандартам X.509. NIST відкрито публікує інформацію, необхідну для виконання цих тестів (наприклад, опис кожного тесту, очікувані результати тесту, і будь-які сертифікати/ключі, списки відкликаних сертифікатів, необхідні для виконання тестів тощо) в режимі он-лайн.

В Україні існують певні пропозиції/ публікацій на цю тему, наприклад, «Тестовий стенд для інтероперабельності електронних цифрових підписів» [45] (Тестовий стенд). У публікації вірно зазначається, що «*одна з основних причин нинішньої відсутності інтероперабельності Національної системи електронних цифрових підписів (НСЕЦП) – прогалини нормативної бази, що регулює технологічну та організаційну складові НСЕЦП*». Для вирішення проблеми пропонується Тестовий стенд, у описі якого також вірно зазначається, що створення тестових стендів (додамо, і криптографічних продуктів взагалі) повинно спиратися *«… на принципи, обов'язкові відповідно до реалізації Загальних критеріїв ІТ-захисту [10], що пов'язано з необхідністю недопущення неякісних реалізацій і заснованих на них претензіях*». Але…

Але подібні пропозиції та відповідні набори тестів, як показано вище на прикладі NIST тестів та відповідних програм, - це завдання, яке вимагає від розробників тестів високої кваліфікації та фахового рівня знань з предмету тестування. По-друге, Тестовий стенд, який

пропонується використовувати для тестування відповідності, сам повинен мати сертифікат відповідності (повинен бути певним еталаном).

Щодо рівня кваліфікації Тестового стенду, який пропонується, то у ньому маємо ототожнення термінів «посилений» та «кваліфікований» сертифікати, про що вже детально було сказано вище у Проблемі №1**.** Так, у описі Тестового стенду зазначено:

«*Загалом вимоги RFC 5280 профілює RFC 3739 (формально RFC 3280, але RFC 5280 замінив RFC 3280) на підтримку посилених сертифікатів. Вимоги RFC 3739 профілює ДСТУ ETSI TS 101 862 на підтримку посилених сертифікатів за визначеннями Директиви 1999/93/ЄС.*»

Але мовою оригіналу маємо «ETSI TS 101 862 V1.2.1 (2001-06) - Technical Specification. Qualified certificate profile». Отже, повинно йтися про «кваліфікований сертифікат» ЄС, який не є еквівалентом «посиленого сертифікату» України. Детальніше про якість та рівень адаптації міжнародних та європейських технічних стандартів (ТК-20) розглянемо далі.

Також, щодо рівня кваліфікації Тестового стенду, який пропонується, то у п. 2.3 «Приклад реалізації стенда» [45] вказується його склад, а саме: «*…використані бібліотеки для роботи з DER-закодованими ASN.1-нотаціями пакету Bouncy Castle, тестові сертифікати згенеровано в операційному середовищі FreeBSD 8.0 за допомогою утиліти OpenSSL*». Зазначені пакети Bouncy Castle та OpenSSL – це пакети «вільного» програмного забезпечення (з відкритим кодом), яке (із тексту ліцензії): «*… надається «як є», без будь-яких гарантій, прямих або непрямих, включаючи, але не обмежуючись гарантії придатності для конкретних цілей (товарного стану, merchantability) …, не несе відповідальності за будь-які претензії, збитки або інші відповідальності …*». До цього слід додати, що версії зазначених пакетів не вказано, а це також є принциповим моментом (наприклад, Bouncy Castle поточна версія 1.46, тобто щонайменше 46-а версія без урахування бета та проміжних версій; з OpenSSL аналогічно). Певно, чітко зрозуміло, що таке «вільне» програмне забезпечення не може бути «еталоном» для оцінювання/ тестування іншого програмного забезпечення (скоріше може бути навпаки). Слід також зазначити, що використання «вільного» програмного забезпечення не є неприпустимим, але обов'язково вимагає додаткового тестування та перевірки правильності, фіксації версії перевіреного пакету, і тільки після цього може використовуватися у безпечних системах.

Також, щодо рівня кваліфікації Тестового стенду, який пропонується, то у 2.2 сказано:

«*Приклад такої недобросовісної реалізації – просте вилучення всіх необхідних полів із сертифіката за допомогою стандартної/незалежної бібліотеки та наявності всіх полів, потрібних за нормативною базою. Цей підхід не проводить детального тестування, наприклад, в кореневому сертифікаті ЦЗО поле підпису, яке має ASN.1 тип BIT STRING згідно з RFC 5280, містить фактичний підпис, закодований як тип OCTET STRING (рис. 3). Верифікація підпису не буде успішною, зважаючи на два додаткових значення, що прикріплено до значення підпису та ідентифікують тип OCTET STRING і його довжину.*

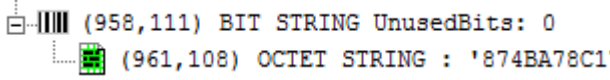


Рис. 3 Закодований підпис ЦЗО».

Для пояснення суті «професійності» зазначеного твердження (*недобросовісної реалізації*) щодо підпису ЦЗО слід надати таку учбову інформацію:

Стандарт RFC 5280:2008 [22] дійсно визначає, що значення підпису кодується як BIT STRING. Але тут же у стандарті сказано, що «*деталі цього процесу/ операції/ оброблення визначаються для кожного з алгоритмів …*» («*The details of this process are specified for each of*

the algorithms …»). Отже, не існує єдиного представлення змісту BIT STRING для усіх алгоритмів.

Стандарт RFC 3279:2002 [23] визначає значення підпису для алгоритму **DSA** (п.2.2.2) так:

При формуванні підпису алгоритм DSA генерує два значення. Ці значення звичайно позначаються як r і s. Щоб легко передати (transfer) ці два значення, як один підпис, вони повинні бути ASN.1 закодовані, використовуючи таку ASN.1 структуру:

```
Dss-Sig-Value  ::=  SEQUENCE  {
        r       INTEGER,
        s       INTEGER  }
```

Стандарт RFC 3279:2002 [23] також визначає значення підпису для алгоритму **ECDSA** (п.2.2.3) так:

При формуванні підпису алгоритм ECDSA генерує два значення. Ці значення звичайно позначаються як r і s. Щоб легко передати (transfer) ці два значення, як один підпис, вони повинні бути ASN.1 закодовані, використовуючи таку ASN.1 структуру:

```
Ecdsa-Sig-Value  ::=  SEQUENCE  {
      r    INTEGER,
      s    INTEGER  }
```

Специфікація «НАЦІОНАЛЬНА СИСТЕМА ЕЛЕКТРОННОГО ЦИФРОВОГО ПІДПИСУ. Технічні специфікації форматів представлення базових об’єктів» визначає значення підпису алгоритму **ДСТУ 4145-2002** (п. 1.3.11.6) як:

Електронний цифровий підпис ДСТУ 4145-2002 – це рядок октетів OCTET STRING (інкапсульовано у полі "signatureValue").

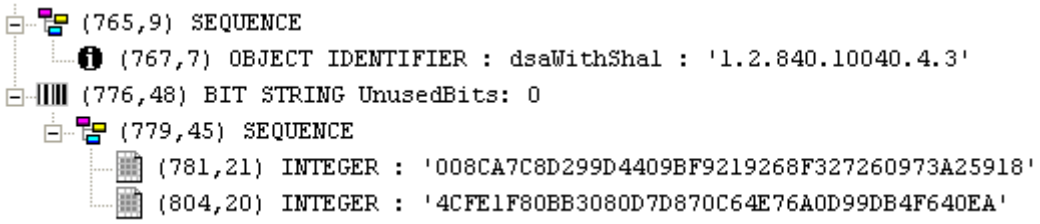
Специфікація «НАЦІОНАЛЬНА СИСТЕМА ЕЛЕКТРОННОГО ЦИФРОВОГО ПІДПИСУ. Технічні специфікації форматів представлення базових об’єктів» визначає значення підпису для алгоритму **ГОСТ 34.310-95** (п. 1.3.11.10) як:

Електронний цифровий підпис згідно ГОСТ 34.310-95 має вигляд (інкапсульовано у полі "signatureValue"):

```
SEQUENCE {
r INTEGER,
s INTEGER
},
```

Таким чином, окрім алгоритму RSA, для якого значення підпису є «чистим» рядком *BIT STRING*, усі інші алгоритми кодують значення підпису або як *SEQUENCE, або як OCTET STRING або інакше, як це передбачено алгоритмом.*

Якщо ж наведених аргументів з цього питання (*недобросовісної реалізації*) недостатньо для переконливості, то пропонуємо звернутися до зазначених вище тестових векторів **NIST** програми PKITS (Public Key Interoperability Test Suite), де зокрема міститься такий тестовий сертифікат (ValidDSASignaturesTest4EE.crt):

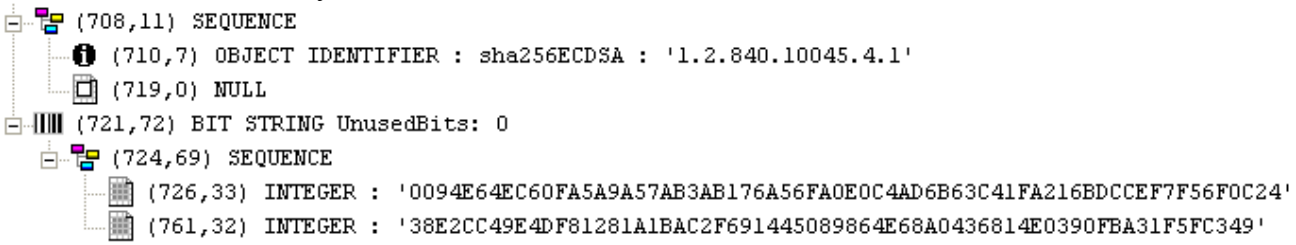


Якщо ж наведених аргументів з цього питання (*недобросовісної реалізації*) недостатньо для переконливості, то пропонуємо також звернутися до сертифікату системи електронних паспортів **Німеччини**:

Видавник сертифікату: CN = csca-germany, SERIALNUMBER = 001, OU = bsi, O = bund, C = DE

Власник сертифікату: CN = DS, SERIALNUMBER = 027, O = Bundesdruckerei GmbH, C = DE

Значення підпису:

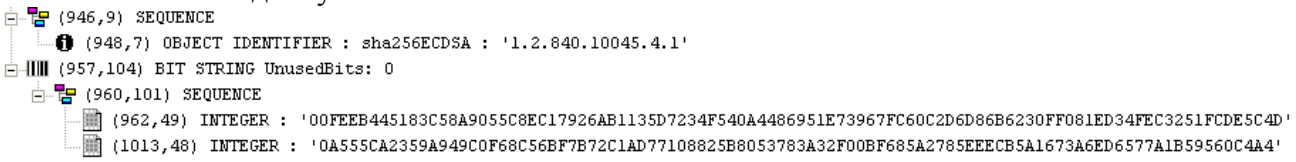


Якщо ж наведених аргументів з цього питання (*недобросовісної реалізації*) недостатньо для переконливості, то пропонуємо також звернутися до кореневого сертифікату системи електронних паспортів **Швейцарії**:

Видавник сертифікату: CN = csca-switzerland-1, OU = Certification Authorities, OU = Services, O = Admin, C = CH

Власник сертифікату: CN = csca-switzerland-1, OU = Certification Authorities, OU = Services, O = Admin, C = CH

Значення підпису:



Висновок: Тестовий стенд, як вважається, не здатний виконувати функції тестування відповідності.

Проблема №3. Низький рівень стандартизації НСЕЦП. Технічні специфікації та адаптовані ДСТУ

Однією з проблем НСЕЦП є недостатній рівень стандартизації в сфері криптографічного захисту інформації (КЗІ).

Існують такі можливості щодо підвищення рівня стандартизації:

1. Адаптація (переклад на державну мову) міжнародних та європейських стандартів.
2. Прийняття міжнародних та європейських стандартів «методом обкладинки» (перекладається назва, «титульний лист», та реєструється як ДСТУ стандарт; зміст стандарту не перекладається, а залишається мовою оригінала).
3. Розроблення та затвердження Технічних специфікацій.

Розроблення та затвердження Технічних специфікацій не гарантує відсутності помилок та недоліків, але термін їх розроблення та затвердження (наказом відповідального державного органу/ установи) складає кілька місяців, що дозволяє оперативно управляти цим процесом.

Прийняття міжнародних та європейських стандартів «методом обкладинки», вважається хорошою практикою, так як є найбільш оперативним заходом. Але слід зазначити, що такий підхід має той недолік «неповного розуміння», тобто один і той же текст англійською мовою може по різному сприйматися різними особами. Цей недолік може бути усунено шляхом видання словників термінів, а при необхідності, уточнень/ рекомендацій на рівні Технічних специфікацій.

Адаптація (переклад на державну мову) міжнародних та європейських технічних стандартів має два суттєвих недоліки:

- Терміни адаптації складають від 2-х (найбільш оптимістична оцінка) до 5-ти років.
- Переклад виконується часто не професійно, а тому користуватися ним неможливо, професійні користувачі звертаються до оригіналу.

Для доказу наведемо деякі приклади непрофесійності адаптації. Слід підкреслити, що приклади надаються, ґрунтуючись як на проектах адаптованих стандартів, так і на вже чинних. На проекти нами свого були надані зауваження та пропозиції щодо внесення змін, але ряд найбільш принципових зауважень було все ж відхилено ТК-20. Тепер, якщо відверто, купувати адаптації такої якості не вбачається доцільним, тому ми в роботі користуємося оригіналами.

Про непрофесійність перекладу (у прикладах)

Розглядаються приклади із низки стандартів ДСТУ ETSI та ДСТУ CWA. Технічний комітет, відповідальний за цей стандарт, – це ТК-20 «Інформаційні технології».

1) Щодо термінів «посилений» та «кваліфікований»

Про відмінність зазначених термінів на законодавчому рівні сказано вище при розгляді Проблеми №1 (Невідповідність законодавства України законодавству ЄС).

У ДСТУ ETSI TS 102 176-1 [24] розділ «Національний вступ» маємо:

«У CWA 14167-1:2004 уведено термін «кваліфікований підпис», визначений у Директиві ЄС як розширений електронний підпис (advanced electronic signature), заснований на посилених сертифікатах (qualified certificate), створених безпечними (надійними) засобами накладання електронних підписів (secure signature creation device). Згідно з Законом України «Про електронний цифровий підпис» від 22 травня 2003 р. №852-IV вважають юридично правомочним (валідним) термін «кваліфікований підпис». »

Таким чином, маємо два різні переклади одного терміну *qualified:*

qualified certificate - посилений сертифікат;

qualified signature - кваліфікований підпис.

Англо-українські словники надають такі переклади «*qualified*»: *кваліфікований, компетентний, придатний, визнаний або правозначущий*. Жодний з можливих перекладів не вказує на те, що можна перекласти це слово як **«***посилений***»**.

У ДСТУ ETSI TS 102 045:2009 [25], у розділі розділ «Національний вступ» маємо аналогічне, а також по тексту:

ДСТУ ETSI	ETSI
Директива 1999/93/ЕС [5] передбачає еквівалентність рукописних підписів, коли	Directive 1999/93/EC [5] provides for the equivalence to handwritten signatures where an

ДСТУ ETSI	ETSI
електронний підпис підтримано посиленими технічними засобами безпеки (стаття 5.1).	electronic signature is supported by enhanced technical security measures (article 5.1).
[9] ETSI TS 101 862 Профіль посиленого сертифіката	[9] ETSI TS 101 862 Qualified Certificate profile
посиленому сертифікаті й згенерованим безпечним засобом створення підпису, зазвичай називають “кваліфікованим електронним підписом”. Як визначено в додатку I, посилений сертифікат має випускати Орган сертифікації, що виконує вимоги Додатку II.	qualified certificate and created on a secure signature creation device, usually known as a "qualified electronic signature." A qualified certificate, as defined in annex I must be issued by a Certificate Authority complying with annex II.

Перелік прикладів можна продовжувати та продовжувати… Це має місце майже у всіх стандартах серій ДСТУ ETSI та ДСТУ CWA Технічного комітету ТК-20.

Як видно з наведених прикладів, йде вибіркова (свідома!?) підміна терміну «кваліфікований» на термін «посилений». Враховуючи принципову відмінність зазначених термінів на законодавчому рівні (див. вище Проблему №1), можна, як вважається, зробити висновок, що переклад не відповідає змісту оригіналу стандартів, тобто не є ідентичним.

2) Щодо перекладу модальних дієслів для позначення рівня вимог

Рівень вимог стандартів відіграє ключову роль у технічних стандартах, так як визначає обов’язковість чи не обов’язковість (опціональність) реалізації певних особливостей/ атрибутів тощо, рекомендується чи не рекомендується (хоча не забороняється) реалізовувати, забороняється чи ні тощо. Без чітких визначень рівня вимог стандартів не може бути й мови про інтероперабельність, бо кожен на свій розсуд буде реалізовувати ту чи іншу підмножину властивостей/ атрибутів/ алгоритмів, забороняти чи дозволяти тощо.

Для визначення рівня вимог стандартів сталою практикою є використання ключових слів та тлумачення модальних дієслів для позначення рівня вимог, відповідно до стандартів RFC, як визначено у RFC 2119 [34].

Мова йде про такі ключові слова (слоформи) стандартів: "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", "OPTIONAL".

Надамо авторський переклад основного змісту RFC 2119 [34]:

1. MUST – Це слово, а також терміни REQUIRED і SHALL використовується для вимог, які є абсолютно необхідними в даній специфікації.
2. MUST NOT – Ця фраза або слова SHALL NOT означають абсолютну заборону в рамках специфікації.
3. SHOULD – Це слово, а також дієслово RECOMMENDED використовується для позначення вимог, від виконання яких можна відмовитися при наявності розумних причин. Однак у разі такої відмови слід пам'ятати про можливі проблеми в результаті відмови і приймати зважене рішення.
4. SHOULD NOT – Ця фраза і дієслово NOT RECOMMENDED використовуються стосовно особливостей або функцій, які допустимі і можуть бути корисними, але можуть викликати проблеми. При реалізації таких опцій слід враховувати можливість виникнення проблем і приймати зважене рішення.
5. MAY – Це слово, а також прикметник OPTIONAL позначають елементи, реалізація яких є необов'язковою. Одні розробники можуть включати такі опції в свою продукцію для розширення можливостей, а інші опускати з метою спрощення. Реалізація, що не включає ту

або іншу опцію, повинна бути готова до роботи з реалізаціями, які використовують цю опцію (можливо спільна робота буде забезпечуватися за рахунок деякого зменшення функціональності). Ті хто включає опцію реалізації, повинні бути готовими (природно, без використання такої опції) до взаємодії з реалізаціями, що таку опцію не підтримують.

6. Рекомендації по використанню. Наведені в цьому документі визначення слід використовувати дуже обережно. Зокрема, необхідно застосовувати їх лише там, де це дійсно диктується вимогами інтероперабельності або для запобігання ситуацій, коли може бути завдано шкоди (наприклад, для обмеження надмірних повторів передачі). Наприклад, такі позначення неприпустимо використовувати для позначення/ трактування переваг однієї реалізації у порівнянні з іншою, якщо це не продиктовано міркуваннями інтероперабельності.

7. Питання безпеки. Розглянуті тут терміни часто використовуються при обговоренні питань безпеки. Відмова від виконання необхідних (MUST) або рекомендованих (SHOULD) вимог або реалізація неприпустимих (MUST NOT)/ не рекомендованих (SHOULD NOT) може істотно впливати на безпеку. Автори документів повинні приділити увагу питанням безпеки, щоб не з'являлися реалізації з невиконаними вимогами або рекомендаціями.

Як вважається, необхідно використовувати такі переклади зазначених ключових слів:
MUST, REQUIRED, SHALL – Необхідно/ Вимагається/ Повинні,
MUST NOT, SHALL NOT – Неприпустимо/ Забороняється,
SHOULD, RECOMMENDED - Рекомендується/ Слідує;
SHOULD NOT, NOT RECOMMENDED – Не рекомендується/ Не слідує;
MAY, OPTIONAL – Можливо/ Опціонально/ Не обов’язково/ Додатково.

Як відомо, переклад модальних дієслів викликає ряд труднощів [42], і тому одним із головних завдань під час адаптації (перекладу) технічних стандартів є максимально чітко, без будь-яких можливостей неоднозначного тлумачення, визначити вимоги, особливо вимоги безпеки.

З точки зору користувача адаптованих стандартів, та з метою максимальної мінімізації можливих неоднозначних тлумачень вимог обов’язковості, слід і навіть необхідно використовувати один і той же термін (словоформу) не тільки в межах одного окремого стандарту, а й у межах усієї серії стандартів певного розділу, наприклад, «Інформаційні технології». Так, з можливих варіантів перекладу «Необхідно/ Вимагається/ Повинні» слід, як вважається, використовувати лише один, наприклад, «Повинні». **Головним чинником для технічних стандартів є не стільки літературність перекладу, скільки чіткість та однозначність тлумачення вимог**.

Враховуючи зазначене, надамо для порівняння текст оригіналу та перекладу із CWA 14167-1:2003 [38]:

Оригінал	Переклад
All security requirements of this CWA are clearly stated and may be: • mandatory (indicated by MUST (NOT) or SHALL (NOT)); • optional (indicated by SHOULD (NOT) or (NOT) RECOMMENDED); • · permitted (MAY or MAY (NOT)).	Усі вимоги безпеки цього стандарту точно заявлено й можуть бути: • обов'язковими (позначено (НЕ) МАЄ або (НЕ) БУДЕ); • опціональними (позначено (НЕ) МАЄ або (НЕ) РЕКОМЕНДОВАНО); • - дозволеними (МОЖЕ або (НЕ) МОЖЕ).

Для того, щоб зрозуміти проблему, закладену таким перекладом, достатньо звернутися до словника української мови, наприклад, «Академічний тлумачний словник (1970—1980)», та ознайомитися з тлумаченням «МАТИ» (http://sum.in.ua/s/maty) та «БУТИ»

(http://sum.in.ua/s/buty), та зробити висновок про те, чи можна їх однозначно тлумачити, а відповідно і реалізовувати для інтероперабельності рішень.

Як слідує з словника, дієслово «БУТИ» жодними чином не вживається на означення вимог обов'язковості (а як майбутній час, існування та інш.). Дієслово «МАТИ» у першу чергу асоціюється з означення того, що комусь належить що-небудь, є його власністю; володіти чимось, посідати щось і так далі - перші 5-ть тлумачень не мають відношення до обов'язковості. І лише на шостому місці ми знаходимо: «6. з інфін. Вживається на означення того, що «*а) (що) повинно щось відбутися, настати. б) (хто) хтось повинен прийти, прибути куди-небудь чи бути присутнім десь. в) (що) повинно щось міститися де-небудь.»*, причому підкреслимо - «з інф.», а тлумачення пояснюється через дієслово «повинен». Отже, як самостійне дієслово «БУТИ», відповідно до «Академічний тлумачний словник (1970—1980)», не може жодними чином відповідати вимозі обов'язковості, і дієслово «МАТИ з інф.» з шостої спроби буде мати вірне тлумачення, як «повинен».

Аналогічно маємо й у ДСТУ ETSI TS 102 176-1 [24]:

Оригінал	**Переклад**
RFC documents use the terms SHALL, SHOULD, MAY, RECOMMENDED in order to allow for interoperability. The same terminology is used in the present document (see RFC 2119 [25]).	У документах RFC вжито терміни БУДЕ, МАЄ БУТИ, МОЖЕ, РЕКОМЕНДОВАНО для досягнення інтероперабельності. Цю термінологію вжито в цьому стандарті (див. RFC 2119 [25]).

Тут додатково ще маємо таку «пропозицію» ТК-20:

SHOULD (Рекомендується/ Слідує) = МАЄ БУТИ.

Технічні спеціалісти при вивченні ДСТУ ETSI TS 102 176-1, з метою встановлення вимог обов'язковості словоформи МАЄ БУТИ, як і у попередньому випадку, звертаються до словника української мови, наприклад, «Академічний тлумачний словник (1970—1980)», але «МАЄ БУТИ» можна знайти лише у та у розділі «БУТИ» (http://sum.in.ua/s/buty), про що йшлося вище, хоча тлумачення словника (*повинно щось відбутися, настати* …) зовсім не відповідає вимозі стандарту «Рекомендується».

Правилом перекладу стандартів повинно бути таке: вживати тільки ті терміни та словоформи, у яких тлумачний словник надає на ПЕРШОМУ місці вірне (з точки зору контексту стандарту) тлумачення.

Можна з упевненістю стверджувати, що майже жоден з технічних спеціалістів, для яких саме і призначається переклад стандарту, не зв'яже МАЄ чи БУДЕ з обов'язковістю, як абсолютно необхідною вимогою. Більшість технічних спеціалістів зв'яже МАЄ БУТИ, як обов'язкову вимогу (повинен), а не рекомендовану.

Якщо ж у стандарті ключові слова вимог обов'язковості не пишуться великими літерами, то отримуємо «вільний» переклад, що виходить поза межі розуміння вимог обов'язковості (ДСТУ CWA 14167-2 [39]):

Переклад	Оригінал	Примітка
FAU_STG.2.3/TOE TSF гарантує, що [призначення: метрика для збереження контрольних звітів] контрольні звіти **буде** підтримано, якщо	FAU_STG.2.3/TOE The TSF **shall** ensure that [assignment: metric for saving audit records] audit records **will be** maintained when the following conditions occur: audit storage exhaustion.	Вимога обов'язковості (**shall**) не врахована. «**буде**» використано для позначення майбутнього часу, а не вимоги «рекомендовано», як у прикладах вище

Переклад	Оригінал	Примітка
виконано умови: вичерпання сховища аудиту		
ATE_FUN.1.2C Плани тестів ідентифікують функції безпеки, які **буде** протестовано, (підлягають тестуванню) і описують мету тестів, які **буде** виконано (підлягають виконанню).	ATE_FUN.1.2C The test plans **shall** identify the security functions **to be** tested and describe the goal of the tests **to be** performed.	Вимога обов'язковості (**shall**) не врахована взагалі. «**буде**» використано для позначення майбутнього часу, а не вимоги «рекомендовано», як у прикладах вище

На зауваження з цього приводу, надані до ТК-20 через НБУ, було отримано категоричну відмову такого змісту: «*Насамперед згадаємо, що неживі сутності нікому нічого не винні/повинні/зобов'язані. Тому слід вживати лише дієслово МАЄ….*». Залишимо проблему перекладу та застосуванню модальних дієслів обов'язковості за професіоналами, фахівцями з лінгвістичної теорії.

Але звернутися до української класики не завадить справі:

Назва твору Т.Г.Шевченка	**Цитата**
«Близнята (1)»	Та проте ці дрібнички спостеріг Степан Мартинович і казав якось у пасіці після читання Тита Лівія, що це недобре: однієї, мовляв, матері діти, то й усе **повинно** бути рівне.
«Близнята (2)»	Це слово мене здивувало: як! у цій мертвій пустелі дерево? І справді вже, якщо воно є, то **повинно** бути святе!
«Мандрівка з приємністю та й не без моралі (1)»	А по правді не **повинно** б так бути. Освіта **повинна** збагачувати, а не обкрадати людське серце.
«Мандрівка з приємністю та й не без моралі (1)»	За півгодини лекція готова, і згідно з умовою інструмент **повинен** бути відчинений.
«Музика (2)» (1855. 15. І. Новопетровський форт)	Але сумна історія, що її мені розказала сердешна Тарасевичівна, **повинна** примусити і німого говорити, і глухого слухати. … На мою думку, твори суто мистецькі не **повинні** описувати картин брудних, хоч це, нажаль, і ввійшло тепер у моду.
«Художник»	— Хто вам підказав, що в мене є його праця? — **Повинна** бути, — сказав він рішуче.
"Щоденник (1857-06)"	Правда — стара, отже **повинна** бути ясна, зрозуміла;
"Щоденник (1857-07)"	8 [липня]. Сьогодні відійшов поштовий човен до Ґурєва. Вітер зюд-вест. У середу або в четвер він **повинен** бути на Стрілецькій Косі, в 15 верствах від Ґурєва.
"Щоденник (1857-08)"	8 [серпня]. На людину, що ниділа, як я, сім літ у голій пустині, кожне, навіть богоспасаєме

Назва твору Т.Г.Шевченка	Цитата
	місто Бєлєбєй (найнікчемніше містечко оренбурзької ґубернії), **повинно** було б зробити приємне враження.
"Щоденник (1857-12)"	Портрет **повинен** бути схожий, бо не подібний до рисунків такого роду.

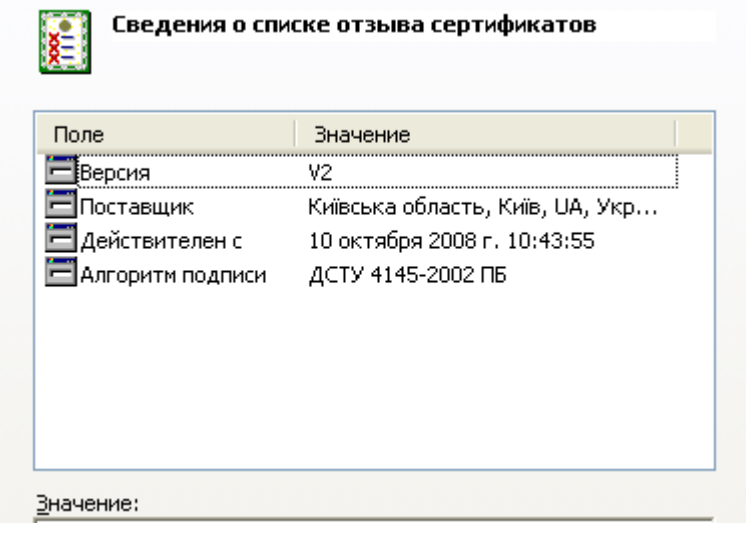
Таким чином, слідуючи Кобзарю, ***дієслово*** *повинен/повинна/повинні* може вживатися і до неживих сутностей. І тому «*…я все ж волію додержувати стилю класичного*» (Т.Г.Шевченко, «Музика (2)).

Категорична відмова з боку ТК-20 щодо вживання дієслова «повинні» тим більш дивує, що ряд стандартів, адаптованих тим же ТК-20, вживають дієслово «повинні» для перекладу MUST, REQUIRED, SHALL:

Стандарт	Цитата
ISO/IEC14888-1:2002 [36]	7.1. Щоб використати специфічний механізм підписування, потрібна специфічна геш-функція. Процес перевіряння **повинен** використовувати тільки цю специфічну геш-функцію.
ISO/IEC14888-1:2002 [36]	9. … Детерміноване свідоцтво **не повинне** передаватися перевірювачу, який також може його обчислювати …
ISO/IEC14888-1:2002 [36]	9.2 … Процес готування повідомлення **повинен** задовольняти одну з двох умов: - Повне повідомлення M **повинне** бути відновлюване для даних M1 і M2;
ДСТУ ISO/IEC 10118-2:2003 [37]	9.2 Вибирання параметрів Параметри L1, L2, та LH геш-функції, наведеної у даному розділі, **повинні** задовольняти: L1 = 3n, L2 = 9n, LH дорівнює 3n.
ДСТУ ISO/IEC 10118-2:2003 [37]	A.6 Мотивування … Ці значення **повинні** мати такі властивості:

Як видно з наведених прикладів, жодних сумнівів щодо подвійного тлумачення вимог не може виникнути – усі вимоги чіткі та однозначні.

Наслідки недотримання (через невірність тлумачення чи інш.) вимог обов'язковості можна показати на прикладі списків блокованих сертифікатів (CRL), що формувалися попередньою версією ЦЗО:



Як видно, CRL має версію 2, поле nextUpdate (наступне оновлення) відсутнє.

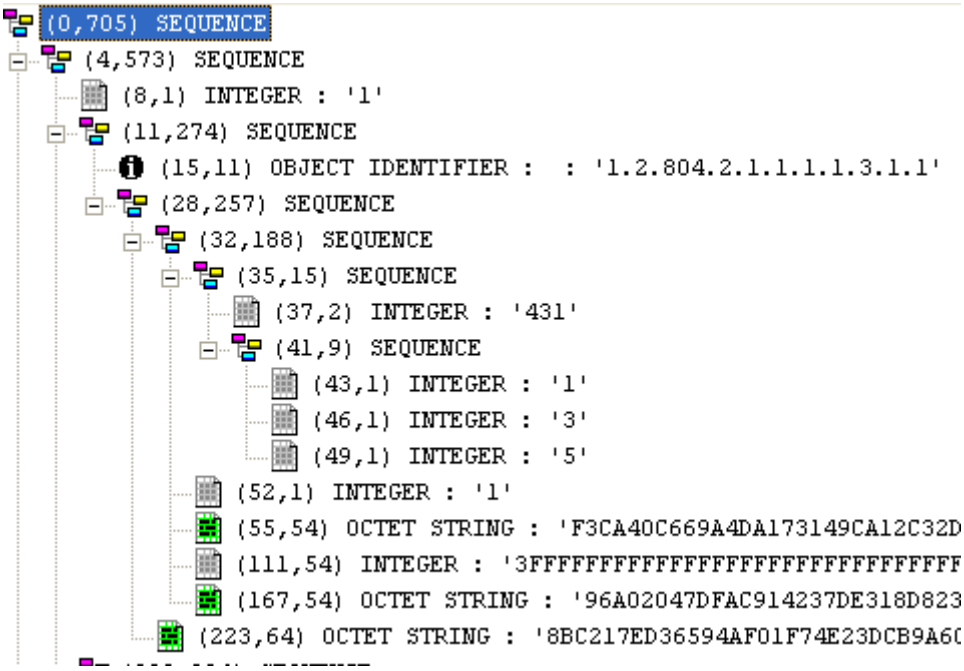
Відповідно до стандарту RFC 5280:2008 [22] (п. 5.1.2.5. Next Update) маємо таку вимогу:

Оригінал	Переклад
Conforming CRL issuers **MUST** include the nextUpdate field in all CRLs. Note that the ASN.1 syntax of TBSCertList describes this field as **OPTIONAL**, which is consistent with the ASN.1 structure defined in [X.509]. The behavior of clients processing CRLs that **omit** nextUpdate is not specified by this profile.	Емітенти CRL, які підпорядковуються правилам, **повинні** включати nextUpdate поле у всіх списках відкликаних сертифікатів. Зверніть увагу, що ASN.1 синтаксис TBSCertList описує це поле як **ОПЦІОНАЛЬНЕ**, що узгоджується з ASN.1структурою, визначеною в [X.509]. Поведінка клієнтів обробки списку відкликаних сертифікатів, що **опускають** nextUpdate, не передбачена цим профілем.

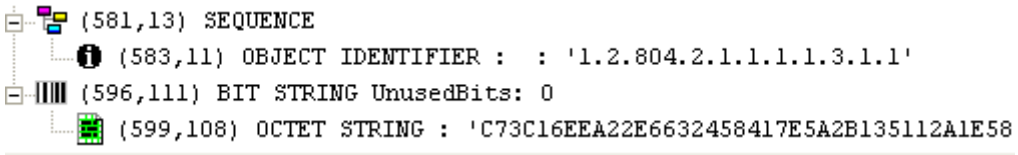
Отже, стандарт вимагає як обов'язкову присутність поля nextUpdate, інакше стандарт не передбачає/ не визначає «*поведінку клієнтів обробки*». Тобто не існує єдиних правил обробки, а отже, не може бути й інтероперабельності.

Другий приклад з тими ж списками блокованих сертифікатів (CRL), що формувалися попередньою версією ЦЗО:

Структура TBSCertList (п. 5.1. CRL Fields стандарту), значення поля signature як AlgorithmIdentifier:



Структура CertificateList (п. 5.1. CRL Fields стандарту), значення поля signatureAlgorithm як AlgorithmIdentifier:



Очевидно, що поле signature не еквівалентне полю signatureAlgorithm (еквівалентність байт-до-байту).

Відповідно до стандарту RFC 5280:2008 [22] (п. 5.1.2.5. Next Update) маємо таку вимогу:

Оригінал	Переклад
5.1.1.2. signatureAlgorithm This field **MUST** contain the same algorithm identifier as the signature field in the sequence tbsCertList (Section 5.1.2.2).	5.1.1.2. signatureAlgorithm Це поле **повинно** містити той же ідентифікатор алгоритму, що і поле signature в послідовності tbsCertList (п.5.1.2.2).
5.1.2.2. Signature This field **MUST** contain the same algorithm identifier as the signatureAlgorithm field in the sequence CertificateList (Section 5.1.1.2).	5.1.2.2. Signature Це поле **повинно** містити той же ідентифікатор алгоритму, що і поле signatureAlgorithm в послідовності CertificateList (п.5.1.1.2).

Зазначена вимога є обов'язковою та означає, що поля signature та signatureAlgorithm, як AlgorithmIdentifier, повинні (обов'язково!) бути еквівалентними.

Через зазначену невідповідність стандарту, оброблення таких CRL списків ЦЗО завершається з помилкою (перевірено для Microsoft Windows XP/2003/7/2008, Sun JRE 1.6, Oracle OAS та інш.). Підкреслимо, причиною неможливості обробити CRL списки ЦЗО попередньої версії було не використання національного алгоритму підпису (ДСТУ 4145) для цих списків, а саме невідповідність стандарту щодо обов'язковості вимог еквівалентності полів signature та signatureAlgorithm.

Є сподівання, що ця проблема тут висвітлена достатньо, або зрозуміти її сутність та шляхи вирішення. Слід лише зазначити, що переклади стандартів містять, як вважається, також термінологічні негаразди щодо суто професійних термінів та значень з інформаційних технологій, криптографії та захисту інформації, але тут це питання не розглядається.

Проблема №4. Низький рівень академічних робіт та досліджень в Україні

Про рівень наукових фундаментальних досліджень з питань, що тут розглядаються, вже можна зробити висновок із попередньо розглянутого рівня кваліфікації Тестового стенду (див. Проблема №2). На жаль, ряд проектів типу *недобросовісної реалізації* виконується, як вважається, в бізнесових цілях особами, які обіймають державні/ наукові посади (адмін.ресурс), що погіршує перспективи інтероперабельності.

Як приклад, див. звернення до Директора Департаменту інформаційно-аналітичного забезпечення процесів оподаткування ДПА України (ttp://forum.sta.gov.ua/posts/list/34228.page), де зазначено, що:

«Інститут кібернетики як один із засновників Національної системи електронних цифрових підписів (НСЕЦП) виконав низку фундаментальних досліджень для вирішення проблем інтероперабельності (функціональної сумісності) об'єктів НСЕЦП щодо основних перешкод у розвитку електронної фіскальної звітності і електронного документообігу. Результати фундаментальних досліджень реалізовано у зареєстрованому центрі

сертифікації ключів (ЦСК) «UPG-PKI» (http://ca.upg.kiev.ua/), який зараз проходить процедуру акредитації.»

Отже, можна тлумачити так, що державна наукова установа, використовуючи «*Результати фундаментальних досліджень»,* проведених в ній за кошти …(?), передала на підставі …(?) ці результати до ТОВ «Юкрейн Проперті Групп» (http://upg.kiev.ua/) для впровадження у зареєстрованому центрі сертифікації для «вирішення» державних проблем інтероперабельності?...

Про «фундаментальність» та рівень, заснований на пакетах «вільного» програмного забезпечення, сказано вище на прикладі Тестового стенду.

Тут зазначимо, що у листі до ДПА пропонується такі заходи (у лапках наведено цитати з листа від імені Інституту кібернетики/ ТОВ «UPG»):

3) Підписування електронних договорів між ДПА і суб'єктами підприємництва:

«*Підписування електронних договорів між ДПА і суб'єктами підприємництва з використанням зручних і звичних засобів MS Office та/або OpenOffice.*»

4) Додати до ДСТУ 4145 алгоритма підпису ще RSA алгоритм (комплект підписів SHA1withRSA):

«*Міжнародний комплект підписів SHA1withRSA реалізовано у ПТК Центрального засвідчувального органу (позитивний експертний висновок 32206929.3КЦД.010.00.1).*

Отже, в НСЕЦП можна застосувати міжнародні комплекти підписів, що значно спростить інтеграцію функцій ЕЦП у сторонні продукти здавання електронної фіскальної звітності, підтримку ЕЦП у шлюзі та внутрішньому електронному документообігу ДПА.»

5) До існуючого шлюзу, що обробляє ДСТУ 4145 додати ще один шлюз для міжнародних ЕЦП:

«*Застосування ДСТУ дозволить вилучити процедуру інтеграції модулів у шлюз, оскільки всі міжнародні ЕЦП, створені іншими криптомодулями, розшифровуватиме єдиний криптомодуль шлюзу. Причому розширення функцій шлюзу згідно з ДСТУ ніяк не вплине на функції обробки ЕЦП, заснованих на ДСТУ 4145.*»

6) Застосувати міжнародні стандарти XML-підписів:

«*Щодо створення потенційної можливості функціонування кількох шлюзів передачі електронних звітів (див. п'яте питання) пропонуємо розпочати застосування міжнародних стандартів XML-підписів і розширюваної мови розмітки звітів XBRL як основного формату електронної фіскальної звітності. Цей формат застосовують у США і більшості країнах-членах ЄС як основний XML-базований формат здачі електронної фіскальної звітності. Застосування єдиного підходу XBRL до формування форм звітності із стандартними XML-підписами дозволить створити альтернативні шлюзи.*»

Розглянемо перший пункт: Підписування електронних договорів між ДПА і суб'єктами підприємництва з використанням «*звичних засобів MS Office та/або OpenOffice»*. Вище вже зазначено, що однією із вирішальних складових європейської програми IDABC інтероперабельності електронного уряду [2] є безпека послуг у цілому, та конфіденційність/ захист персональних даних зокрема.

Коротко, - на цей час не існує міжнародних/європейських стандартів щодо цифрового підпису/ шифрування документів MS Office (Word, Excel …). Існують міжнародні та європейські стандарти лише для XML-документів та PDF-документів. Отже, «*використання зручних і звичних засобів»* не завжди є прийнятним, якщо при цьому зовсім не розглядається безпека послуг у цілому, та захист персональних даних зокрема.

Розглянемо другий пункт: Додати комплект підписів SHA1withRSA.

По-перше, заміна одного криптографічного алгоритму іншим не може автоматично «*значно спростити інтеграцію функцій ЕЦП у продукти здавання електронної фіскальної звітності»*. Це залежить, як зазначено вище, від Організаційної та Семантичної

інтероперабельності, від вірного вирішення *Технічної інтероперабельності,* що включає в себе [2], зокрема:

- відкриті інтерфейси,
- інтеграцію даних і допоміжне програмне забезпечення (middleware),
- представлення даних (data presentation) і обмін даними,
- безпеку послуг у цілому, та конфіденційність персональних даних зокрема.

Більш детально про істинні проблеми «*інтеграції функцій ЕЦП»* буде зазначено далі. Якщо коротко, то заміна одного комплекту підпису на інший не може забезпечити інтероперабельність.

По-друге, комплект підпису SHA1withRSA з 01.01.2012 не рекомендовано (з міркувань безпеки) для використання [24].

Розглянемо третій пункт: До існуючого шлюзу, що обробляє ДСТУ 4145 додати ще один для міжнародних ЕЦП. При цьому нібито «*всі міжнародні ЕЦП, створені іншими криптомодулями, розшифровуватиме єдиний криптомодуль шлюзу».* Така впевненість може бути тільки у випадку незнання істинних причин несумісності різних крипто модулів, що розглянуто далі у цій публікації (див. Проблему № 5 «Низький рівень стандартизації НСЕЦП. Несумісність різних крипто модулів»). Тут лише зазначимо, що перехід на інші алгоритми підпису, навіть на міжнародні, не забезпечить автоматично сумісності криптографічних модулів різних виробників, особливо це стосується, на жаль, українських виробників.

Розглянемо четвертий пункт: Застосування міжнародних стандартів XML-підписів. Така позиція є вірною, особливо тому, що ДПА використовує документи формату XML. Але, слід зазначити, що з точки зору реалізації, XML-підпис та шифрування є найбільш складним завданням із такого ряду завдань:

- підпис у форматі PKCS#7/CMS (синтаксис криптографічних повідомлень);
- підпис PDF-документів;
- підпис XML-документів.

На сьогоднішній день в України регламентовано (на рівні технічних специфікацій) лише підпис у форматі PKCS#7/CMS (синтаксис криптографічних повідомлень), який і використовується ДПА та іншими учасниками НСЕЦП, і який до того ж є найпростішим з точки зору реалізації та інтероперабельності.

Підпис PDF-документів стандартизовано міжнародними та європейськими стандартами (в Україні не регламентовано). За реалізацією – це більш складне завдання, так як вимагає: а) реалізації підпису у форматі PKCS#7/CMS, б) реалізацію стандарту ISO 32000-1:2008 [26] з урахуванням серії стандартів ETSI TS 102 778 [27-33].

Підпис у форматі XML також не регламентовано в Україні. Для інтероперабельності не допоможуть тут і міжнародні та європейські стандарти, бо вони описують універсальний формат з безліччю варіантів та необов'язкових (опціональних) елементів/ компонентів підпису/ шифрування. Ці необов'язкові елементи по-різному (мається на увазі обсяг числа таких елементів) можуть реалізуватися різними розробниками. Отже, без технічної чіткої регламентації формату XML підпису та шифрування його впровадження не допоможе інтероперабельності.

Таким чином, як вважається, наведений приклад не є зразком дійсно наукових фундаментальних досліджень щодо НСЕЦП.

Проблема № 5. Низький рівень стандартизації НСЕЦП. Несумісність різних крипто модулів

Керуючись основними *принципами* інтероперабельності (відкритість інтерфейсів, слідування стандартам) [2], розглянемо **головні причини** неінтероперабельності НСЕЦП,

використовуючи для прикладів систему електронної звітності ДПА України (аналогічний стан і в інших системах електронної звітності, наприклад, Пенсійного фонду України та інш.):

1) «Багатовекторність» технічних специфікацій України

Під «багатовекторністю» будемо розуміти кілька варіантів реалізації, закладених у технічних специфікаціях для однієї задачі.

Так, у технічній специфікації форматів об'єктів [40] фактично «подвоєно» кількість алгоритмів ДСТУ 4145-2002 – маємо не два алгоритми (поліноміальний базис та оптимальний нормальний базис), а фактично чотири:

Для формату Little-Endian (при визначенні параметрів еліптичної кривої у сертифікаті):
поліноміальний базис 1.2.804.2.1.1.1.1.3.1.1
оптимальний нормальний базис 1.2.804.2.1.1.1.1.3.1.2
Для формату Big-Endian (при визначенні параметрів еліптичної кривої у сертифікаті):
поліноміальний базис 1.2.804.2.1.1.1.1.3.1.1.1.1
оптимальний нормальний базис 1.2.804.2.1.1.1.1.3.1.2.1.1

Відрізняються ці пари алгоритмів лише форматом кодування параметрів точки еліптичної кривої у сертифікаті. Тобто ніякого сенсу, з точки зору безпеки та криптографічних властивостей, це не дає, а лише ускладнює реалізацію (тобто збільшує терміни, вартість розробки тощо) та збільшує ймовірність несумісності.

Автору невідомо, щоб існував міжнародний/європейський стандарт чи технічна специфікація, де б один криптографічний алгоритм мав два об'єктних ідентифікатори для різних форматів кодувань параметрів алгоритму (Big-Endian, Little-Endian), а тим паче для кодування лише частини параметрів.

Слід зазначити, що така «багатовекторність» не є головною причиною, але будь-яке необґрунтоване, як конче необхідне для цілей безпеки/ криптографічної стійкості, ускладнення реалізації алгоритму - це:

а) додаткові витрати на створення та подальше підтримання/ обслуговування алгоритму/ криптографічного засобу;

б) збільшення можливих помилок та несумісності, особливо при недостатньому рівні тестування, про йшлося вище (див. Проблема №2. Низький рівень стандартизації НСЕЦП. Тестування відповідності у сфері КЗІ).

Висновок: При створенні технічних специфікацій не слід намагатися «догодити» усім розробникам, навіть «впливовим», а слід керуватися виключно стандартами та кращою міжнародною і європейською практикою.

2) Створення криптографічних об'єктів типу «2в1» (два в одному)

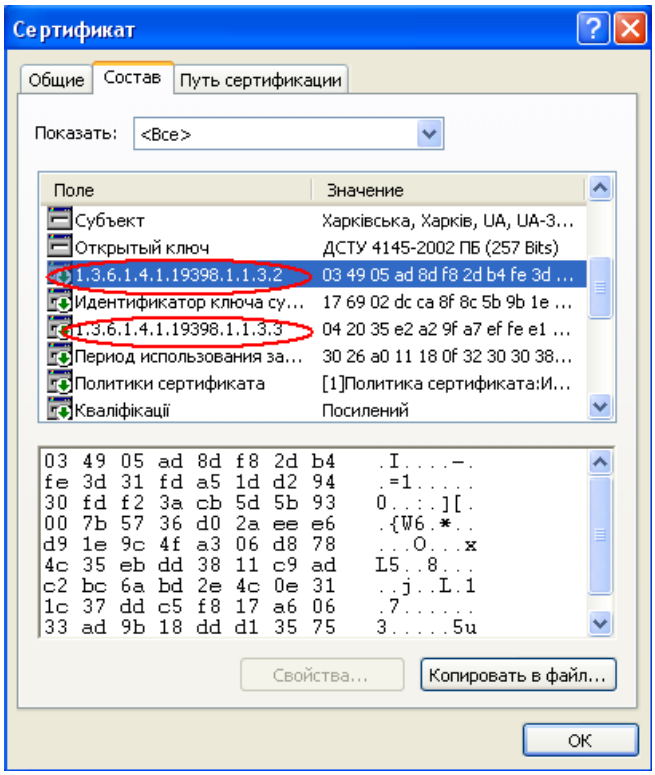
Стандартами X.509 визначається структура сертифікату відкритого ключа, обов'язкові та опціональні атрибути і розширення сертифікату. Стандартами X.509 дозволяється додавати до структури сертифікату власні розширення розробника, не визначені стандартами. У цьому випадку, звичайно, зміст таких розширень невідомий нікому (що це і що з ним робити, як обробляти …?), окрім розробника, а тому такі розширення взагалі не повинні розглядатися для інтероперабельних систем.

Але… Кілька розробників в Україні, та відповідно й кілька акредитованих центрів сертифікації ключів (АЦСК), створили та використовують X.509 сертифікати цифрового підпису з власними розширеннями, не визначеними стандартами чи офіційними/ публічними

специфікаціями (причому ці розширення суттєво впливають на оброблення цифрового підпису/шифрування), що є порушенням принципів інтероперабельності.

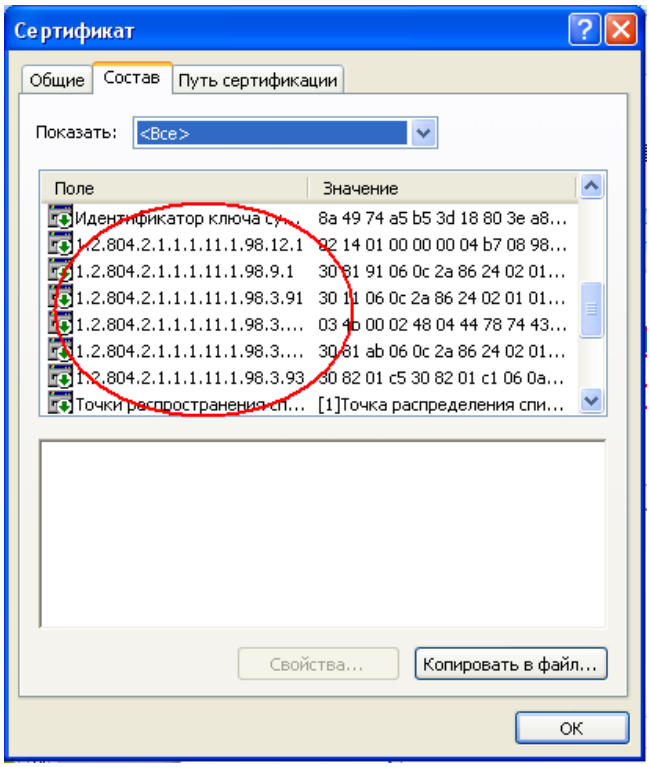
Окремі розробники розмістили в цих розширеннях ключі шифрування, тобто в одному сертифікаті фактично міститься два відкритих ключа – один для підпису, другий для шифрування (через це назвемо такі об'єкти «2в1»), наприклад:

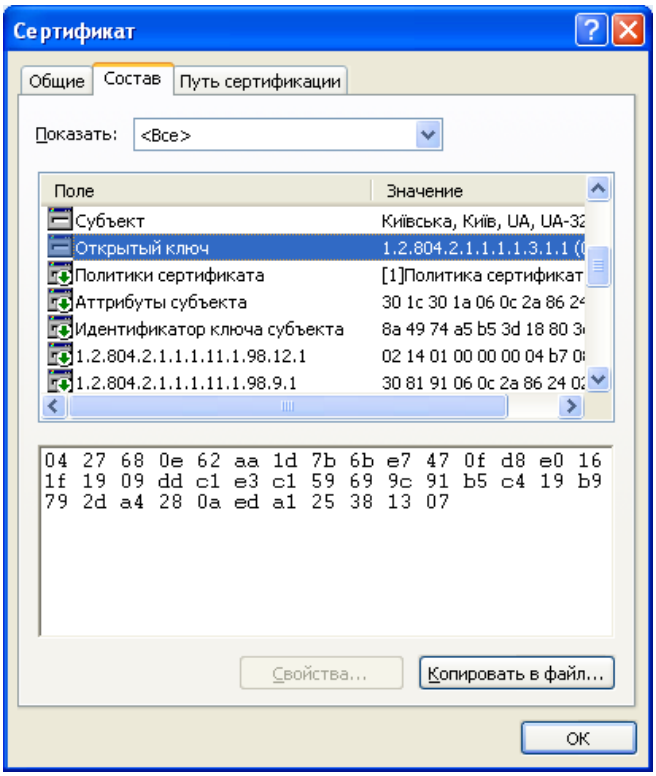
Приклад №1. Сертифікат АЦСК в Україні:



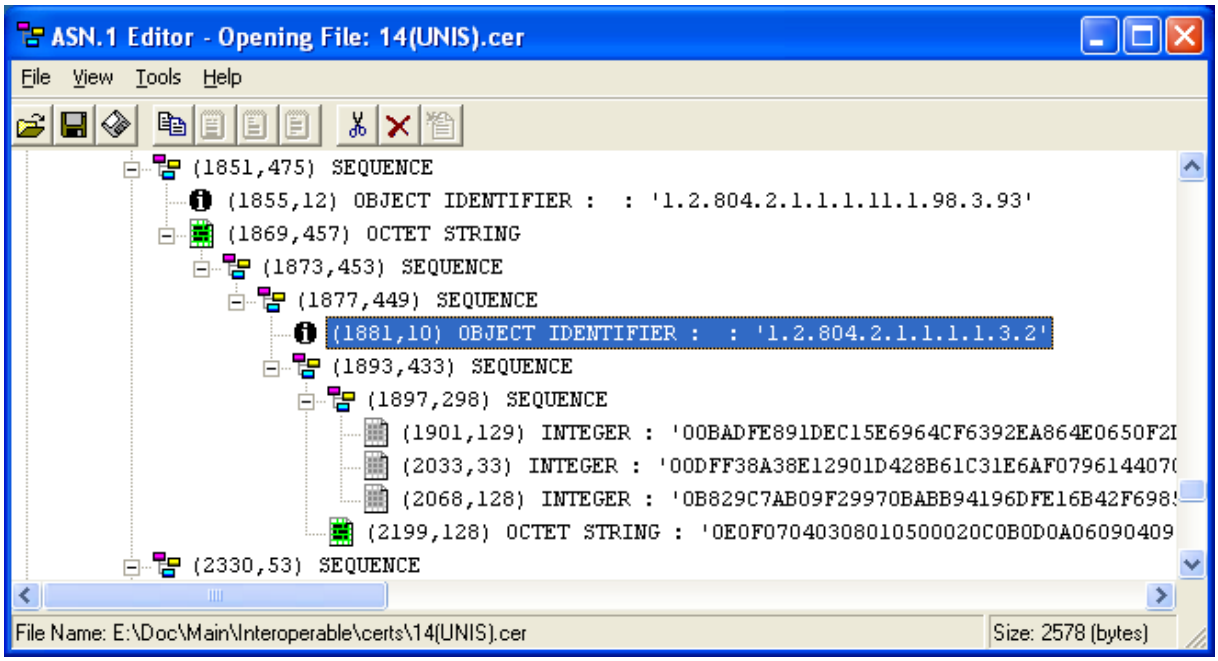
У цьому сертифікаті присутні розширення з такими об'єктними ідентифікаторами (OID): 1.3.6.1.4.1.19398.1.1.3.2, 1.3.6.1.4.1.19398.1.1.3.3, , які не описані в стандартах чи специфікаціях, не зареєстровані належним чином, а отже, зміст та призначення цих розширень невідомі.

Приклад №2. Сертифікат АЦСК в Україні:





Наприклад, розширення з OID =1.2.804.2.1.1.1.11.1.98.3.93 має таке значення:



Так як OID = 1.2.804.2.1.1.1.1.3.2 означає алгоритм підпису ГОСТ 34.310-95 з геш функцією ГОСТ 34.311-95 (Gost34310WithGost34311) (див. [40]), то можемо припустити, що в розширенні розміщено параметри алгоритму підпису ГОСТ 34.310-95 з геш функцією ГОСТ 34.311-95. Але ж сам сертифікат видано на відкритий ключ, що має алгоритм з OID = 1.2.804.2.1.1.1.1.3.1.1, тобто ДСТУ 4145-2002 та довжину ключа 307 бітів. Отже, один сертифікат відкритого ключа містить два ключа – ДСТ 4145 та ГОСТ 34.310. Призначення та правила оброблення другого ключа невідомі.

Розглянемо наскільки така практика «2в1» відповідає стандартам X.509. Відповідно до стандартів X.509, зокрема RFC 5280:2008 [22]:

1. Сертифікат засвідчує та реєструє один відкритий ключ, що міститься в полі PublicKey.
2. ЦСК повинен перевіряти унікальність відкритого ключа та те, що цей ключ пов'язаний з відповідним йому особистим ключем (PrivateKey).
3. Сертифікат у полі KeyUsage містить інформацію про використання (призначення) ключа PublicKey – для підпису, шифрування тощо.
4. Рекомендується термін ключа шифрування встановлювати не більше одного року, а термін ключа підпису 1-2 роки.

У разі розміщення в розширенні сертифікату другого ключа (ключів), ці положення стандарту не можуть бути застосовані для другого ключа (ключів), а отже цей додатковий ключ(і) не може вважатися чинним(и).

Висновок: Наявність невідомих розширень, які вимагають оброблення під час формування підпису/ перевіряння підпису/ зашифрування/ розшифрування є неприпустимим для інтероперабельних систем, незалежно від алгоритму відкритого ключа сертифікату (ДСТУ 4145, RSA, ECDSA тощо).

3) Відсутність технічних специфікацій шифрування

Однією з причин, що породила хибну практику «2в1», про яку йшлося в попередньому пункті, була відсутність технічних специфікацій з шифрування на відкритих ключах ДСТУ

4145-2002. На даний момент така технічна специфікація вже створена, хоча ще не впроваджена в практичну діяльність АЦСК, а отже, проблема все ще існує.

Відсутність технічних специфікацій з шифрування у поєднанні з практикою «2в1» призвела до того, що кожен АЦСК надавав користувачам власні («власного бачення/ розуміння») алгоритми шифрування/ розшифрування та відповідні бібліотеки функцій власного формату (з інтерфейсами «власного бачення/ розуміння»).

Замість того, щоб вжити заходів щодо формулювання загальних правил для усіх розробників шляхом створення відповідних технічних специфікацій інтероперабельності, в системах електронної звітності (наприклад, ДПА України), вирішили проблему інакше – шляхом створення відомчого шлюзу, який поєднає у собі безліч бібліотек «власного бачення/ розуміння» розробників АЦСК.

Підтримувати та супроводжувати на будь-якому шлюзі безліч бібліотек різних виробників – це, м'яко кажучи, не найкраща практика.

Висновок: Впровадити Технічні специфікації з шифрування на відкритих ключах ДСТУ 4145-2002. Не може й не повинно бути необхідності у будь-яких шлюзах, якщо усі АЦСК будуть відповідати єдиним стандартам шифрування. Без цього інтероперабельність неможлива, незалежно від алгоритму ключа/ сертифікату (ДСТУ 4145, RSA, ECDSA тощо).

4) Відсутність технічних специфікацій сховища ключів/ сертифікатів

Сховище ключів та сертифікатів призначається для безпечного зберігання закритих ключів та відповідних сертифікатів на носіях інформації. При цьому під сховищем ключів розуміють саме файлове сховище (програмна реалізація криптографічного модулю), а не будь яку апаратну реалізацію криптографічного засобу. Питання взаємодії із сховищами ключів та сертифікатів є невід'ємною складовою інтероперабельності.

При створенні відомчого шлюзу (приклад ДПА) додатково, для того, щоб була можливість виконувати крипто операції в обидва напрямки (клієнт-шлюз, шлюз-клієнт), необхідно на шлюзі також мати **особисті ключі** кожного з АЦСК (нагадаємо – через несумісність форматів «2в1» та алгоритмів шифрування).

Намагаючись хоч будь-як впорядкувати роботу з різними бібліотеками різних розробників, ДПА затверджує Уніфікований формат транспортного повідомлення [44], у додатку 3 якого міститься «Специфікація криптографічних функцій». Але…

У параметрах зазначених криптографічних функцій є такі: «*Буфер з секретним ключем*», «*Буфер з сертифікатом*», але відсутні функції пошуку та/чи отримання цих «*буферів*» (звідки вони будуть братися?).

Таким чином, на сьогодні в системі електронної звітності ДПА необхідно:

1. Отримати від усіх розробників АЦСК бібліотеки криптографічних функцій та інтегрувати їх у шлюз.
2. Створити біля (у зоні доступу) шлюзу безпечні сховища ключів та сертифікатів (за кількістю АЦСК, чи типів АЦСК за кількістю розробників).
3. Якимось чином (не визначеним у Специфікаціях) отримувати «*Буфер з секретним ключем*» для передавання його криптографічній функції на виконання операції.

Отже, ДПА зробило перший крок до уніфікації, створивши Специфікації, але залишило без уніфікації сховища ключів та сертифікатів, як і процедури/ функції звернення до них.

Існує лише один міжнародний стандарт, що застосовується для сховища ключів та сертифікатів у вигляді файлу (файлове сховище) – це стандарт PKCS#12 [40], який для внутрішнього зберігання ключів використовує PKCS#8, а також PKCS#5, PKCS#9 та інші стандарти.

Для апаратних , програмно-апаратних криптографічних модулів стандарти форматів сховищ відсутні; стандартами вимагається забезпечення рівня безпеки зберігання та функціонування. Для таких засобів стандартизуються відкриті інтерфейси (див. наступний пункт, стандарт PKCS#11). Слід зазначити, що для апаратних , програмно-апаратних засобів сховище може бути створене у будь-якому форматі, який розробник вважає кращим та безпечнішим. Але у цьому випадку повинні стандартизуватися відкриті інтерфейси.

Окрім сховищ типу PKCS#12 існує також сховища Microsoft Windows, формат яких, певно, визначено специфікаціями Microsoft, які не мають статусу стандартів.

Стандарт PKCS#12 використовується для резервування (резервного копіювання) ключів та сертифікатів з наступним імпортом їх до програмних, програмно-апаратних, апаратних криптографічних модулів.

Використання сховищ формату PKCS#12 поширено у Java JVM, Linux/Unix операційних системах, але його головне призначення виплаває з назви стандарту «Personal Information Exchange Syntax» (Синтаксис обміну особистою інформацією), тобто для обміну. Слід зазначити такі недоліки/ слабкості файлових сховищ:

- Відсутність захисту від копіювання;
- Відсутність захисту від атаки «грубої сили» (brute force attack);
- Під час виконання криптографічних операцій особистий/секретний ключ залишає «пристрій», тобто сховище, та передається до функцій виконання операції, чим може бути порушена безпека ключа.

Через це файлове сховище використовується, як правило (з міркувань безпеки):

- для резервування (резервного копіювання) чи обміну (експорту/ імпорту) ключів та сертифікатів;
- на захищених комп'ютерах (у захищеному середовищі), де є достатні гарантії безпечного використання ключів.

В окремих публікаціях щодо інтероперабельності НСЦЕП також обговорюється проблема сховища ключів, наприклад [45], але висновки, що зроблені, є невірними. Так у зазначеній публікації пропонується:

«*Более того, для внутренней интероперабельности на территории Украины, не говоря уже о кроссертификации, необходим стандартный формат личного ключа, например, на основе де-факто стандарта PKCS#8*»

По-перше, до кроссертифікації питання формату особистого ключа не має жодного відношення -на крос сертифікацію подається X.509 сертифікат, а не особистий ключ (який, нагадаємо, не повинен залишати криптографічний модуль).

По-друге, використовуючи формат PKCS#8 особистого ключа на рівні прикладних програм (звітності тощо), нібито з метою інтероперабельності, ми тим самим передаємо на цей рівень (прикладний) безпосередньо сам особистий/ секретний ключ (його значення), чим втрачаємо будь-який контроль над ключем. Отже, автоматично такий криптографічний модуль не може бути «Безпечним засобом створення підпису», як це вимагається Директивою 1999/93/ЕС. Такий хибний, як вважається, підхід бачимо і у рішенні ДПА, де вимагається «*Буфер з секретним ключем*». Про те, що інтероперабельність і безпека по суті є антагоністичними вимогами йшлося на початку цієї публікації. Часто, і навіть занадто часто, на практиці при створенні автоматизованих систем будується певний концепт функціональності та інтероперабельності без жодної думки щодо безпеки. Для довідки: безпечні засоби створення підпису ніколи (за жодних обставин) не повинні передавати на рівень прикладного застосування (поза межі криптографічного модулю чи програмного, чи програмно-апаратного, чи апаратного) значення особистого ключа. Операційна взаємодія здійснюється через так звані дескриптори (ідентифікатори/ аліаси/ псевдоніми) закритого

ключа, а сам ключ, його значення, при цьому не залишає межі криптографічного модуля. Тільки за цих умов криптографічний модуль вважається безпечним.

По-третє, формат PKCS#8 особистого ключа не містить жодної інформації про відповідний йому (цьому ключу) відкритий ключ та сертифікат, а отже немає можливості встановити однозначну відповідність особистого ключа та його відкритого ключа/ сертифіката. У цьому випадку навіть стандартизація формату особистого ключа та сертифіката не надасть інтероперабельності, бо механізм відповідності особистого ключа сертифікату залишається невизначеним. Отже, знову розробники АЦСК «на власний розсуд/ розуміння» будуть реалізовувати його, і, звичайно ж, кожен по-своєму, бо відповідного стандарту не існує.

Для інформації: Microsoft реалізує по-своєму механізм відповідності ключа сертифікату; у стандарті PKCS#12 це реалізовано через аліаси/ псевдоніми (aliases); у стандарті PKCS#11 це реалізовано через внутрішній ідентифікатор об'єкту ID, який є спільним для групи об'єктів закритий ключ - відкритий ключ - сертифікат.

Висновок: Без уніфікації та стандартизації сховища ключів та сертифікатів, інтероперабельність проблематична, а без стандартизації процедур/ функцій звернення до ключів та виконання крипто операцій, інтероперабельність неможлива незалежно від алгоритму ключа/ сертифікату (ДСТУ 4145, RSA, ECDSA тощо).

5) Відсутність відкритість інтерфейсів

Для програмних, програмно-апаратних криптографічних модулів стандартизуються відкриті інтерфейси, визначені стандартом PKCS#11 [40]. Жоден відомий виробник подібних засобів (не України) не постачає їх на ринок, не забезпечивши реалізацію стандарту PKCS#11.

Відкритість інтерфейсів досягається також шляхом створення так званих Криптографічних сервіс-провайдерів (Cryptographic Service Provider, CSP), створення яких, як правило, базується на інтерфейсі PKCS#11, тобто як надбудова над PKCS#11. У цьому випадку для інтероперабельності можуть використовуватися як PKCS#11, так і більш високий рівень інтеграції – CSP.

Існує де-факто два типи реалізацій CSP: Microsoft CSP, та Java CSP (JCA/JCE). Майже усі відомі виробники криптографічних засобів (не України) постачають їх на ринок, забезпечивши реалізацію Microsoft CSP та Java CSP (JCA/JCE). А так як іноземні виробники відповідно реалізують міжнародні алгоритми підпису/ шифрування, то саме цим і досягається інтероперабельність для міжнародних алгоритмів. Тобто, проблема не у використанні конкретно ДСТУ 4145 чи RSA (див. висновок у публікації [47]), а проблема у рівні якості виробу щодо відповідності стандартам, зокрема, щодо відкритих інтерфейсів. Тобто, якщо реалізувати RSA чи інші міжнародні алгоритми, але з тим же рівнем «якості» («на власний розсуд/ розуміння» розробника), то отримаємо такий же рівень неоперабельності, який мається сьогодні з ДСТУ 4145.

Слід підкреслити, що відкриті інтерфейси, визначені стандартом PKCS#11, можуть застосовуватися не тільки для апаратних/ програмно-апаратних, а й для чисто програмних криптографічних модулів.

Висновок: Без уніфікації та стандартизації відкритих інтерфейсів криптографічних модулів інтероперабельність неможлива незалежно від алгоритму ключа/ сертифікату (ДСТУ 4145, RSA, ECDSA тощо).

Так чи інакше, в основі зазначених вище причин неінтероперабельності НСЕЦП лежить недотримання головних принців інтероперабельності [2]: слідування стандартам та відкритість інтерфейсів. Отже, незалежно від того, який конкретно алгоритм/ набір

алгоритмів, міжнародних/ національних тощо, буде використовуватися для ЕЦП, - без усунення зазначених причин, НСЕЦП і надалі буде залишатися неінтероперабельною.

Загальний висновок: Для практичного досягнення інтероперабельності НСЕЦП України вважається за необхідне таке:

1. Внести зміни до Закону України «Про електронний цифровий підпис», привівши його у відповідність до Директиви 1999/93/ЕС;
2. Створити відкриті методики та набори тестових векторів для тестування відповідності у сфері криптографічного захисту інформації; регламентувати процес та критерії оцінювання;
3. Активізувати роботу щодо створення технічних специфікацій, які ґрунтуються на міжнародних та європейських стандартах;
4. Підняти на якісно новий рівень роботи з адаптації міжнародних/ європейських стандартів з інформаційних технологій та криптографічного захисту;
5. Невідкладно вжити практичних заходів щодо (технічно-технологічного) приєднання та впровадження в Україні європейської програми IDABC (Інтероперабельне надання послуг європейського електронного уряду державним адміністраціям, бізнесу та громадянам) шляхом впровадження відповідних стандартів;
6. Заборонити розроблення/ створення будь-яких відомчих технічних специфікацій тими державними установами/ відомствами, які не є відповідальними за цей напрямок та не мають відповідної кваліфікації.

P.S. Ця публікація є оціночним судженням, переконаннями, критичною оцінкою певних фактів і недоліків поточного стану НСЕЦП України, які виражені як суб'єктивна думка автора, відповідно до Закону України «Про інформацію» та Конституції України щодо права на свободу думки і слова, на вільне вираження своїх поглядів і переконань. Ця публікація не має на меті приниження честі та гідності, чи ділової репутації будь-якої особи, а піклується виключно інтересами забезпечення професійності інтероперабельності НСЕЦП України.

Перелік літератури

1. CEC. Commission of the European Communities, Council Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs (1991) - Official Journal L 122 , 17/05/1991 P. 0042 - 0046;
2. IDABC. Enterprise & Industry DG, European Interoperability Framework for panEuropean e-government services, version 1.0, - Luxembourg: Office for Official Publications of the European Communities, Brussels, 2004, ISBN 92-894-8389-X;
3. SAGA. Standards and Architectures for e-government Applications – KBSt1 Publication Series, ISSN 0179-7263, Volume 59, December 2003;
4. FIPS PUB 140-2 Security Requirements for Cryptographic Modules - Information Technology Laboratory National Institute of Standards and Technology (NIST), Gaithersburg, MD 20899-8900, Issued May 25, 2001;
5. ISO/IEC 15408-1:2009 -- Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 1: Introduction and general model - 2009-12-03;
6. ISO/IEC 15408-2:2008 -- Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 2: Security functional components - 2008-08-19;
7. ISO/IEC 15408-3:2008 -- Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 3: Security assurance components - 2008-08-19;
8. ДСТУ CWA 14365-2:2009 - Настанова щодо використання електронних підписів. Частина 2. Профіль захисту для програмних засобів створення підпису (CWA 14365-2:2004, IDT), - Дата введення в дію: 01.07.2011;

[11] KBSt (Koordinierungs- und Beratungsstelle der Bundesregierung für Informationstechnik in der Bundesverwaltung) - Co-ordinating and Advisory Agency of the Federal Government for Information Technology in the Federal Administration (KBSt), Координаційне та консультативне агентство федерального уряду з інформаційних технологій у Федеральній адміністрації/ Федеральному управлінні.

9. ДСТУ CWA 14167-3: 2008 - Криптографічний модуль для послуг генерування ключів провайдером послуг сертифікації. Профіль захисту CMCKG-PP (CWA 14167-3:2004, IDT), - Дата введення в дію: 01.01.2010;

10. ДСТУ-П CWA 14172-5:2008 - Настанова Європейської Ініціативи стандартизації електронних цифрових підписів з оцінювання відповідності. Частина 5. Безпечні засоби створення підпису (CWA 14172-5:2004, IDT), - Дата введення в дію: 01.01.2009;

11. ДСТУ-П CWA 14172-6:2008 - Настанова Європейської Ініціативи стандартизації електронних цифрових підписів з оцінювання відповідності. Частина 6. Засіб створення підписів, що підтримує підписи, крім кваліфікованих (CWA 14172-6:2004, IDT), - Дата введення в дію: 01.01.2009;

12. ДСТУ-П CWA 14172-7:2008 - Настанова Європейської Ініціативи стандартизації електронних цифрових підписів з оцінювання відповідності. Частина 7. Криптографічні модулі, використовувані провайдерами послуг сертифікації для операцій підписування та послуг генерування ключів (CWA 14172-7:2004, IDT), - Дата введення в дію: 01.01.2009;

13. ДСТУ CWA 14355:2009 - Настанова щодо реалізації безпечних засобів створення підписів (CWA 14355:2004, IDT), - Дата введення в дію: 01.07.2011;

14. CWA 14169:2004 - Secure Signature-Creation Devices "EAL 4+";

15. CWA 14170:2004 - Security requirements for signature creation applications;

16. Директива 1999/93/ЄС Європейського парламенту та Ради «Про систему електронних підписів, що застосовується в межах Співтовариства» від 13 грудня 1999 року. – Офіційний переклад (http://zakon.rada.gov.ua/cgi-bin/laws/main.cgi?nreg=994_240);

17. Закон України «Про електронний цифровий підпис» від 22 травня 2003 року №852-IV - Відомості Верховної Ради України (ВВР), 2003, N 36, ст.276);

18. NIST. Public Key Interoperability Test Suite (PKITS). Certification Path Validation, Version 1.0, September 2, 2004;

19. NIST. Path Discovery Test Suite - Version 0.1.1, June 3, 2005;

20. The Advanced Encryption Standard Algorithm Validation Suite (AESAVS) November 15, 2002;

21. The XTS-AES Validation System (XTSVS) - Updated: March 2, 2011; Previously Updated: August 30, 2010; Original: March 31, 2010;

22. RFC 5280:2008 – Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile;

23. RFC 3279:2002 - Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile;

24. ДСТУ ETSI TS 102 176-1:2009 - Електронні підписи та інфраструктури (ESI). Алгоритми та параметри безпечних електронних підписів. Частина 1. Геш-функції й асиметричні алгоритми (ETSI TS 102 176-1:2007, IDT);

25. ДСТУ ETSI TS 102 045:2009 - Електронні підписи та інфраструктури (ESI). Політика підписів для розширеної бізнес-моделі (ETSI TR 102 045 V1.1.1 (2003-03), IDT);

26. ISO 32000-1:2008 "Document management -- Portable document format -- Part 1: PDF 1.7",

27. ETSI TS 102 778 V1.1.1 (2009-04) - Technical Specification - Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; CMS Profile based on ISO 32000-1

28. ETSI TS 102 778-1 V1.1.1 (2009-07) - Technical Specification - Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 1: PAdES Overview - a framework document for PAdES

29. ETSI TS 102 778-2 V1.2.1 (2009-07) - Technical Specification - Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 2: PAdES Basic - Profile based on ISO 32000-1

30. ETSI TS 102 778-3 V1.2.1 (2010-07) - Technical Specification - Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 3: PAdES Enhanced - PAdES-BES and PAdES-EPES Profiles

31. ETSI TS 102 778-4 V1.1.2 (2009-12) - Technical Specification - Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 4: PAdES Long Term - PAdES-LTV Profile

32. ETSI TS 102 778-5 V1.1.2 (2009-12) - Technical Specification - Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 5: PAdES for XML Content - Profiles for XAdES signatures

33. ETSI TS 102 778-6 V1.1.1 (2010-07) - Technical Specification - Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 6: Visual Representations of Electronic Signatures

34. RFC 2119:1997 – Key words for use in RFCs to Indicate Requirement Levels;

35. ДСТУ ETSI TS 102 176-1:2004 Електронні підписи та інфраструктури (ESI). Алгоритми та параметри безпечних електронних підписів. Частина 1. Геш-функції й асиметричні алгоритми;

36. ДСТУ ISO/IEC14888-1:2002 – Методи захисту. Цифрові підписи з доповненням. Частина 1. Загальні положення (ISO/IEC14888-1:1998, IDT);

37. ДСТУ ISO/IEC 10118-2:2003 Інформаційні технології. Методи захисту. Геш-функції. Частина 2. Геш-функції з використовуванням n-бітового блокового шифру;

38. CWA 14167-1:2003 - Security Requirements for Trustworthy Systems Managing. Certificates for Electronic Signatures - Part 1: System Security Requirements;

39. ДСТУ CWA 14167-2:2004 Криптографічний модуль для операцій підписання провайдером послуг сертифікації з резервним копіюванням - Профіль захисту CMCSOB PP;

40. PKCS #11 v2.20: Cryptographic Token Interface Standard - RSA Laboratories, 28 June 2004;
41. PKCS 12 v1.0: Personal Information Exchange Syntax – RSA Laboratories, June 24, 1999;
42. К. С. Борзілова. ЛІНГВІСТИЧНА ТЕОРІЯ. ТРУДНОЩІ ПРИ ПЕРЕКЛАДІ МОДАЛЬНИХ ДІЄСЛІВ - Вісник ЛНУ імені Тараса Шевченка № 9 (220), Ч. ІІІ, 2011;
43. НСЕЦП. Технічні специфікації форматів представлення базових об'єктів - Наказ Департаменту спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України, Державного департаменту з питань зв'язку та інформатизації Міністерства транспорту та зв'язку України від 11 вересня 2006 р. № 99 /166;
44. Уніфікований формат транспортного повідомлення при інформаційній взаємодії платників податків і податкових органів в електронному вигляді телекомунікаційними каналами зв'язку з використанням електронного цифрового підпису, наказ ДПА України від 12.07.2010 № 499;
45. Мелащенко Андрій Олегович, Перевозчикова Ольга Леонідівна. Тестовий стенд для інтероперабельності електронних цифрових підписів, - Наукові записки Києво-Могилянської академії. – К.: Видавничий дім „Києво-Могилянська академія", 2010. – С. 54-61;
46. А. О. Мелащенко, О. Л. Перевозчикова, РОЛЬ КОМПЛЕКТОВ ПОДПИСЕЙ В КВАЛИФИЦИРОВАННОЙ ИНФРАСТРУКТУРЕ ОТРЫТЫХ КЛЮЧЕЙ - Математичне та комп'ютерне моделювання, Серія: Технічні науки, Випуск 3, - 2010, стор. 138-154;
47. Мелащенко Андрей Олегович, Перевозчикова Ольга Леонидовна - Проблемы интероперабельности Национальной системы электронных цифровых подписей, - Кибернетика и системный анализ. – 2009. – № 3. – Р 55-63.

Ця публікація підписана цифровим підписом автора (martyn@itsway.kiev.ua). При цьому використано ключ RSA, хоча може бути підписано будь-яким іншим, у тому числі ДСТУ 4145, чи ECDSA, чи інш.

Мартыненко Сергей Васильевич

Подписано цифровой подписью: Мартыненко Сергей Васильевич
DN: cn=Мартыненко Сергей Васильевич, st=Киев, l=Киев, sn=Мартыненко, givenName=Сергей Васильевич, title=Директор технічний, email=martyn@itsway.kiev.ua, serialNumber=38
Дата: 2011.12.09 14:34:21 +02'00'