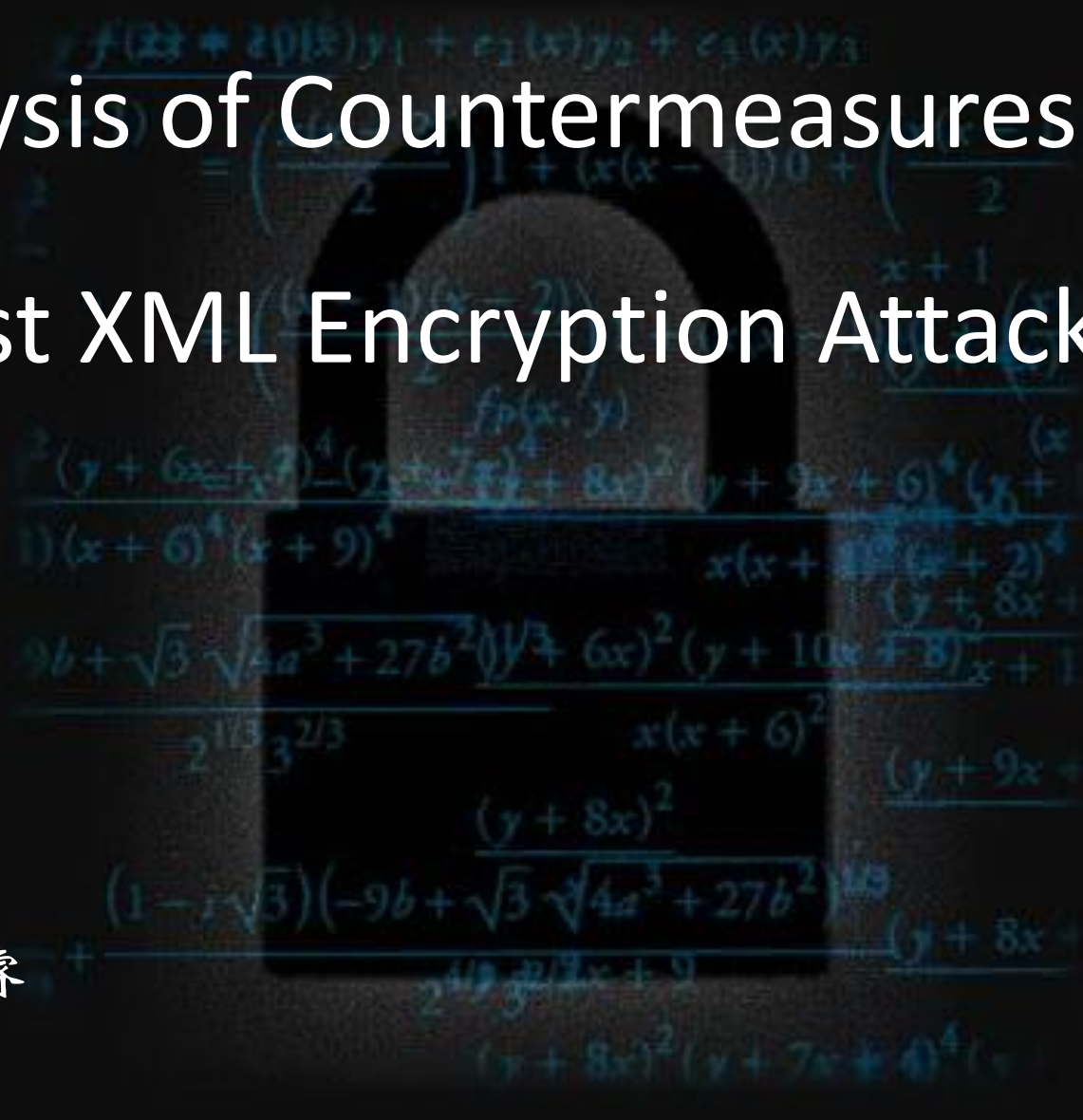
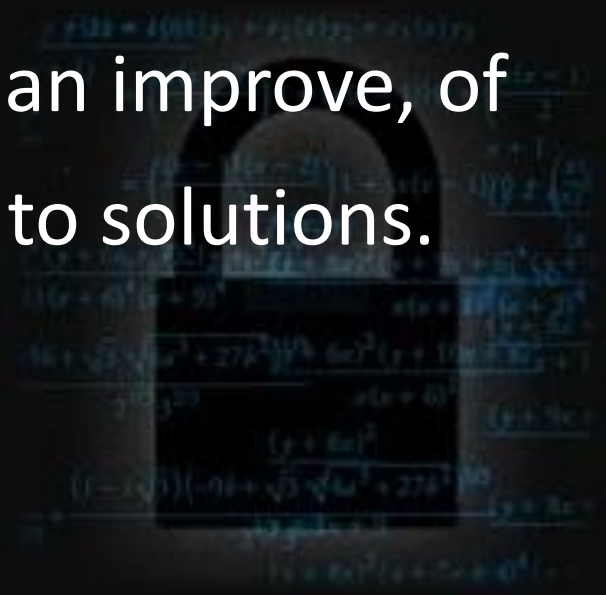
Analysis of Countermeasures against XML Encryption Attacks



Group 9

张超 张颖婕 刘睿
王岱鑫 张天扬

Contempory standard of XML-Enc is vulnerable to attacks, and XML-Sig does nothing help. All this mess is an inevitable result of CBC-mode weakness. Only a complete change, rather than improve, of the existing standard leads to solutions.



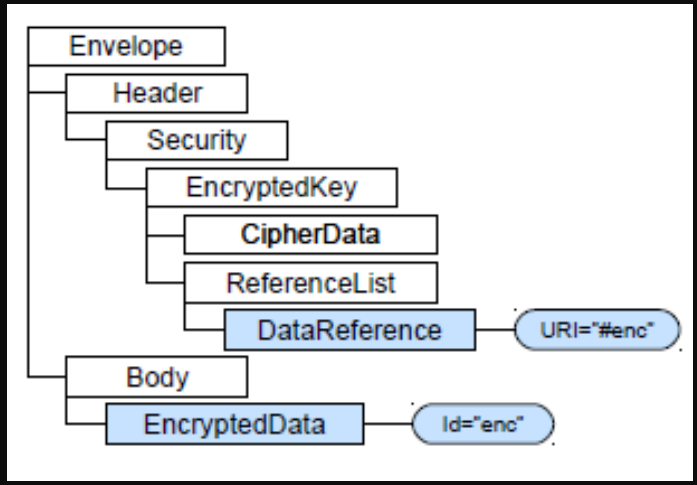
Preliminaries

- Extensible Markup Language (XML)
- XML Encryption (XML-Enc)
- XML Signature (XML-Sig)
- Simple Object Access Protocol (SOAP)

Backgrounds

- Padding Oracle Attack, 2002
- XML-Sig Element Wrapping Attacks, 2005
- XML-Enc Wrapping Attacks, 2011
- XML-Enc Attacks, 2011

Process of XML-Enc



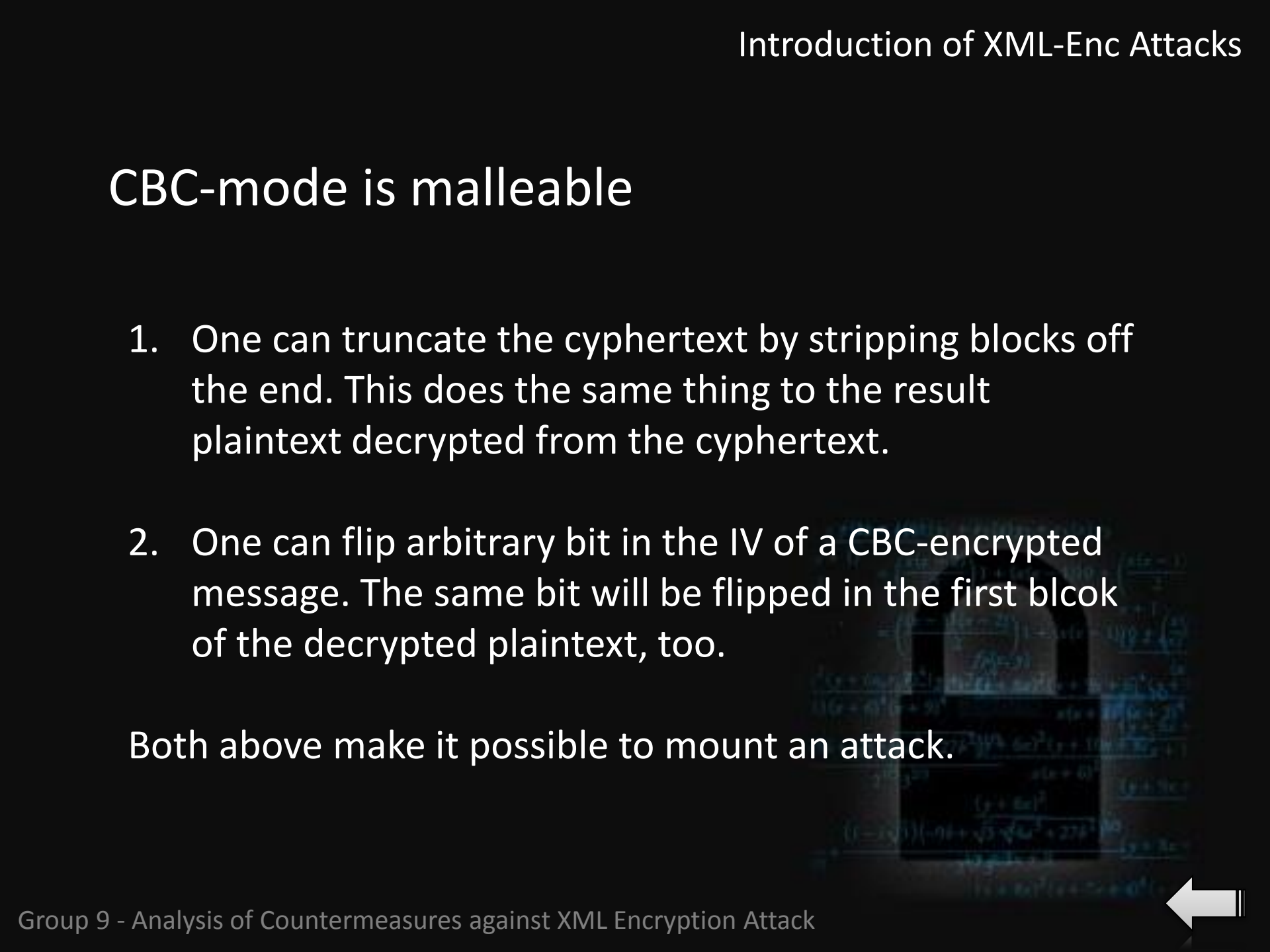
1. Plaintext is divided into blocks
2. Xor plaintext block 1 with IV
3. Encrypt result using AES or 3DES, get cyphertext block 1
4. Xor plaintext block i with cypher block i-1, and encrypt resulting cyphertext block I
5. IV and all cyphertext blocks compose EncryptedData

CBC-mode is malleable

1. One can truncate the cyphertext by stripping blocks off the end. This does the same thing to the result plaintext decrypted from the cyphertext.

2. One can flip arbitrary bit in the IV of a CBC-encrypted message. The same bit will be flipped in the first blcok of the decrypted plaintext, too.

Both above make it possible to mount an attack.



How to break XML-Enc ?

1. Trucncate the cyphertext.
2. Tweak the padding.
3. Squish the bug(s).
4. Learn the last byte of the block.
5. Set padding to 0x01.

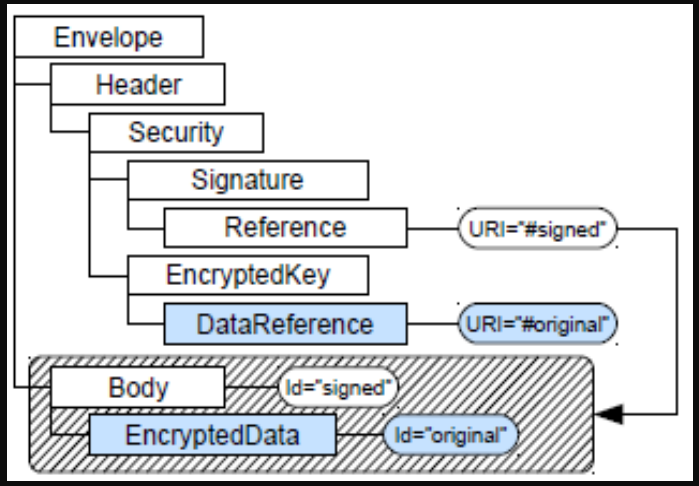
How to break XML-Enc ? (Cont'd)

6. Observe the character set, e.g. ASCII.
7. Flip some bits of IV and get oracle.
8. Loop step7 until learn everything.
9. After figure out the first block, others can be done in the same way.

It cannot work so easily if …

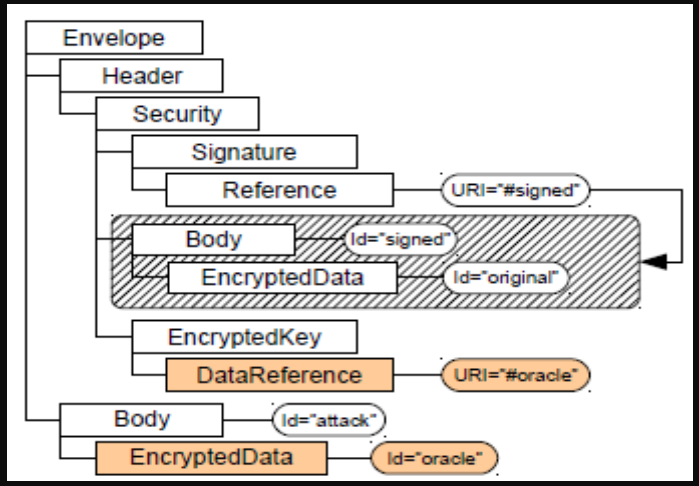
- One is not able to modify cyphertext as he or she likes.
- No meaningful response can be got from server.
- Never endure continuous sending of invalid messages from client.
- Authenticated encryption instead of CBC.

Why and how XML-Sig ?



- XML-Sig is used to ensure message integrity
- Types : pulic-key and secret-key
- Structures : enveloping, enveloped, detached
- For detached XML-Sig, we have id-based and Xpath-based referencing.

XML-Sig Wrapping Attacks



1. Copy signed SOAP body to security head.

2. Rename id.

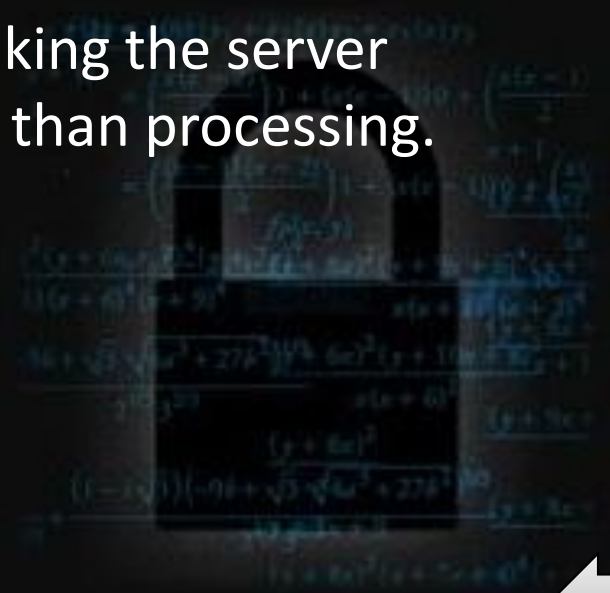
3. Now the content of newly copying body can be modified without invalidating signature.

However, by appling Xpath, a query language of selecting nodes from an XML document, to XML-Sig referencing, this kind of attacks may not work.

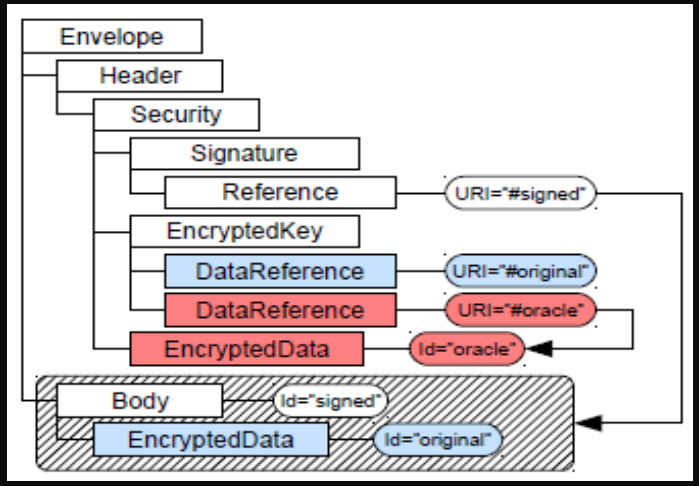
But that is not the end

Although XML-Sig can ensure message integrity, it only cares what will be processed.

XML-Enc Attacks only tends to get plaintext from cyphertext, so the necessary step is making the server decrypting modified cyphertext, rather than processing.



XML-Enc Wrapping Attacks



1. Copy the encrypted payload to any unsigned document part.

2. Create a new DataReference element pointing to the newly created EncryptedData element.

3. Now the content of the new EncryptedData can be modified.

In this way, the server is forced to decrypt chosen ciphertext of attackers.

Difficulties of XML-Sig

In XML-Enc wrapping attack scenarios, modified cyphertext is placed somewhere unsigned, e.g. in SOAP head. The server have to decrypt it to figure out what the message is. And this is enough for attackers to get useful oracles.

To conclude, current standard of XML-Sig does nothing to XML-Enc attacks.

Unifying error messages

To mount XML Enc attacks, one needs to get response from the server about modified cyphertext sent before. What if no response ?

To mount XML Enc attacks, one needs to get response from the server about modified cyphertext sent before. What if no response ?

1. Meaningful error messages are necessary for developers of client-side applications.

2. Besides error messages, there are additional ways to detecting what type of error a certain request message triggered. As an example, measuring the time consumed until a (unified) SOAP fault message arrives may already indicate the level in the application stack at which the error occurred.

Limit reuse of session keys

In the process of attack, one need to use the same symmetric key for each server request. The server can revoke session keys in this situation.

In the process of attack, one need to use the same symmetric key for each server request. The server can revoke session keys in this situation.

1. It requires synchronization among servers in a cluster.

2. Sometimes a few bits are also important. Revocation is always too late.

3. It does not work in any situation. Sometimes the same symmetric key is used for encrypting all messages for the entire duration of a session.

1-way sending

For message proctols based on XML-Enc, sending the encrypted message over with 1-way-SSL can make it impossible for attackers to get original cyphertext.

It is not efficient in all situations. In fact, this strategy requires unendurable cost of changing current web structure.

Backlisting of clients

It is evidently useless. The attack can be applied from any client terminal in any place. Strategies based on forbidding repeating request from certain clients will never work.

Current situation

Actually, XML-Enc attacks are proved to be efficient on nearly all major web services framework :

- Apache Axis 2
- RedHat Jboss
- IBM WebSphere
- Microsoft .NET
- And more…

The solution is to change XML-Enc, rather than servers.

Revolution of XML-Enc standard

- A new encrypting and decrypting restriction

Stop blindly decrypt everything that is encrypted in the message. This in turn requires forbidding encryption of any part of XML document as one likes.

Revolution of XML-Enc standard (Cont'd)

- Authenticated encryption instead of CBC

Since the development of chosen cyphertext attacks, CBC-mode keeps being vulnerable in various situations. XML-Enc standard forms in 2002, and seems to remain unchanged until now. It is essential to change the standard and apply more secure encryption method, e.g. GCM.

References

1. Tibor Jager, Juraj Somorovsky : How to Break XML Encryption
2. Juraj Somorovsky, Jorg Schwenk : Technical Analysis of Countermeasures against Attack on XML Encryption
3. Serge Vaunenay : Security Flaws Induced by CBC Padding
4. Micheal McIntosh, Paula Austel : XML Signature Wrapping Attacks and Countermeasures
5. Sebastian Gajek, Meiko Jensen, Lijun Liao, Jorg Schwenk : Analysis of Signature Wrapping Attacks and Countermeasures
6. Donald Eastlake, Joseph Reagle, Takeshi Imamura, Blair Dillaway, and Ed Simon : XML Encryption Syntax and Processing. W3C Recommendation, 2002.
7. Donald Eastlake, Joseph Reagle, David Solo, Frederick Hirsch, and Thomas Roessler : XML Signature Syntax and Processing (Second Edition). W3C Recommendation, 2008.
8. Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, and Henrik Frystyk Nielsen : SOAP Version 1.2. W3C Recommendation, 2003.

Thank You !

&

Questions

Group 9

张超 张颖婕 刘睿
王岱鑫 张天扬