Webology, Volume 18, Number 5, 2021
ISSN: 1735-188X
DOI: 10.29121/WEB/V18I5/57

A DES, AES, DSS, and RSA-Based Security System for Protecting Sensitive Information During Communication and Providing Fast, Reliable File Identification

Prabhdeep Singh[1], Vikas Tripathi[2], Durgaprasad Gangodkar[3], Dibyahash Bordoloi[4]

[1]Department of Computer Science & Engineering,Graphic Era Deemed to be University,
Dehradun, Uttarakhand India, 248002
[2]Department of Computer Science & Engineering,Graphic Era Deemed to be University,
Dehradun, Uttarakhand India, 248002
[3]Department of Computer Science & Engineering,Graphic Era Deemed to be University,
Dehradun, Uttarakhand India, 248002
[4]Head of the Department, Department of Computer Science & Engineering,Graphic Era Hill
University, Dehradun, Uttarakhand India, 248002

ABSTRACT

Web-based financial, insurance, and other corporate applications have made data security a top priority. Confidentiality, endpoint authentication, message integrity, and no repudiation are only some of the security criteria that must be met by such apps. Data security standards establish XML vocabularies and processing rules for encryption/decryption and signatures/validation of documents. In this research, we offer a safe architecture for exchanging sensitive documents, which involves the transfer of files between a sender (the server) and a receiver (the client) through a central gateway using web services applications. The only files that need to be sent over the internet are those that contain sensitive information, thus it's important to design a system with built-in security measures like digital signatures. Encryption/decryption, signature/validations, and web services have all seen their fair share of algorithmic innovation, implementation, and coding.

Keywords: Confidentiality, Authentication, Algorithms and Methods

INTRODUCTION

The sender and the recipient both use the central gateway, which is a Java programme, to transfer a sensitive document. Using this programme, we can provide various security services to computers on the outside. Encryption, decryption, signatures, and validation are all part of the security services provided by a well-designed system. Sender and gateway conduct encryption and signature generation; receiver and gateway conduct decryption and validation. Systems security has been improved by the established usage of symmetric and asymmetric keys. The system contains a predetermined list of email addresses to whom the document or documents should be sent. When sending a sensitive file, it is best practise to encrypt it with the appropriate signatures and then keep

it on the centralised gateway.
To verify and decode the encrypted XML file, the recipient must first visit the central gateway, which is sent to through the confirmation email. Automatic deletion of older encrypted files from the gateway database occurs in sync with the creation of newer files in terms of both time and date. All the recipients get their encrypted files and may decode them at their convenience.

Further, Web Services Security offers message-level security for web services. It's a kind of communication that allows for the incorporation of WS security. The

WS communications may be made private with the right precautions. Web Service Security Encrypt (WSSE) policy files govern WS-Security. [1] When using WS-Security to gain entry to a protected web service, a policy file must be produced and linked to the service's management interface. This has to be in sync with the web service's policy file. When sending data to a web service that expects encrypted input, the corresponding control file must encrypt the data before sending it. In the midst of all the system's transactions, the web service is crucial. onfidential papers may be safely sent from sender to recipient in this fashion.

Validating the identity of the correct authorised user requires efficient file authentication. Identity validation is at stake here. Digital certificates are used to verify identities in open e-commerce systems. Therefore, digital signatures and correct validation of the signers' signatures are used for authentication on both the sender's and the recipient's ends. Document integrity is also verified so that the recipient can tell whether the message's contents have been tampered with. In order to verify the authenticity of a file, digital signatures are widely utilised. Here, non-repudiation is crucial because it guarantees that if integrity and authentication are guaranteed, then the non-repudiation condition will also be met. The five pillars of security—confidentiality, availability, authentication, integrity, and non-repudiation—are all satisfied via the use of these techniques and algorithms (DES, AES, DSA, and RSA, to name a few).

CRYPTOGRAPHIC IDEA

It is essential to use cryptographic methods for constructing a safe document transfer system. The fundamental motivation for cryptography was the need to conceal communication. Common components of cryptographic operations include keys (both public and private), key pair generators, key manufacturers, key storage, and cryptographic algorithms. There are two main types of cryptographic systems: those that use a single key, known as a symmetric-key system, and those that use two keys, known as public-key systems. The idea of transferring information in an encrypted form is widely used in the financial sector and other businesses where privacy and security are paramount. It's more than just a way to make sure only the right people can decipher your data; it's a method for meeting a wide range of security needs.

The following are the four prerequisites for message-level security:

First, confidentiality guarantees that the information sent remains private. No one except the intended recipient will be able to read it. To provide information encryption is used to provide privacy. Second, the message's integrity guarantees that the recipient will notice if the message's contents have been tampered with. In order to verify the authenticity of a file, digital signatures are widely utilised. Third, authentication, refers to the process of confirming someone's identification. Digital certificates are used to verify identities in open e-commerce systems. Lastly, non-repudiation ensures that the criteria may be met provided integrity and authenticity are guaranteed. [2] For the purpose stated above to be achieved, it is imperative that this article adhere to these four primary and essential conditions. Assuming these objectives are met, the system as a whole should be safe. Without achieving all of these objectives, the system cannot be considered secure. To do this, all that is required is the use of globally acknowledged standards when transmitting any private document to multiple banks, huge enterprises, IT firms, and small businesses. To illustrate the use of such a system, consider the following scenario: "For example, Receiver desires to order and pay for a book from Sender using the mutually trusted payment method Zip Pay" (Prepaid card as issued by Meta-Bank). The sender makes up an order form that has the book's title, price, and payment details. As of right now, the sender wishes to sign all of his/her data, but will encrypt his/her account credentials for zip Pay alone. The sender sends this to the recipient, who verifies the book title and price, signs the form twice, and then submits the order, along with the sender's payment details, to Zip Pay for processing. Zip Pay must be aware that both the cypher data version of the encrypted information and the plain data form are required to confirm the signatures of the sender and the recipient. If the signed material is encrypted at some point after the signature procedure, then the signature cannot be validated without first decrypting the encrypted sections. Decryption procedures are provided by the transform, with just the signed and encrypted data being decrypted at first.

If there is concern that a third party could encrypt parts of the signature, the signer might include this transform in the transform sequence, i.e. the canonical XML. There is an intranet/client side (the sender), an internet/server side (the receiver), and a gateway (Zip Pay) in this scenario. This approach has several applications in the commercial world, including banks, corporations, government organisations, and more. Third, a number of options for implementing the secure transmission. Several of the following methods are discussed as potential answers to the problem of system security. For the purpose of developing the java application, i.e. the web service, these solutions have been worked on in both a practical and theoretical manner.

The first solution is that the system flows safely when the WS-Security services are used correctly. Encryption, digital signatures, decryption, and verifications are some of the security services available. These services must be utilised in the correct format, which implies that the order of starting and end positions must be respected. Implementing the system effectively necessitates the employment of suitable algorithms, and it is essential that these algorithms adequately support the aforementioned security services. Failure to correct the problem will result in improper operation of the system. In order to provide this safety

Moreover, Sun has created certain preset APIs for addressing the system's issue in the most effective manner possible by using specific algorithms.

The second option is to set up the necessary tools for building web services and constructing basic apps. There has to be integration between the client and the gateway. The gateway's implementation is the most crucial and significant aspect of the system, since it's what ultimately allows everything to work. The secure papers will never make it through the system without this Java programme.

Configuring the gateway with the aid of security services is essential for the system's overall security. A client is a simple programme that allows any user to interact with a gateway. Digital signatures and encryption standards ensure that all four processes (encrypt, decrypt, sign, and verify) are given a document that can be trusted. The development of security services relies on the correct application of keys. Symmetric keys are used for encryption and decryption inside a Java programme, whereas public and private keys are used during the signing and validation processes. While the addition of a third party, in this case a gateway (GW), increases the complexity of the system's implementation, doing so correctly ensures the system's overall security. The ultimate answer is to set up the server (on the receiver side) and the gateway such that they may talk to each other without any problems. Java programming language is used to code the predefined APIs that are used to support the java applications. Determine the programming language that will be used to create the system and the necessary tools to run it. Given the complexity of these applications, only visual tools can meet the needs of the system.

Outlined in the figure below are the four procedures that make up the system's design: encryption, decryption, signatures, and validations.

CRYPTOGRAPHICAL FUNCTIONS

Different cryptographic operations are represented by the following notation:
Product = Algorithmic Function [Input | Key Type | Owner]
Mathematical Expression (1).
The terms "Function", "Input", and "Output" are required here. More particular information about a function may be provided by adjusting other parameters. A "Function" is defined by its inputs and its outputs. A certain "algorithm" may be used to implement the "Function." A "key" is used to operate most features. The "type" and "owner" of a key may be specified."
Here are some examples of the variables that may be used:
Purpose: "E, D, H, S...
"RSA, DES, AES, DSA, etc., are all algorithms.
Labels: open, closed, session, etc. Where "E," "D," "H," and "S" stand for "Encryption," "Decryption," "Hash," and "Digital Signatures," respectively. [3]
Ways in which the methods may be put to use
Methods for Protecting Information "Three methods are listed below in which the encryption must function: First, use symmetric encryption only for protecting the file: For encrypting a file, only one session key should be used, since it is expected that same key will also be used when decrypting the file. This implies that the symmetric key is the only one needed for both encryption and decryption. Therefore, the key must be loaded throughout the operation and safeguarded while kept separately from the encrypted file. This method is used to plan the overall system. Two-factor authentication (two-factor auth) is a method of authenticating a user's identity and access to a resource, such as a file, using a secret key or password. The document contains both the encrypted session key and the encrypted data. Using the public asymmetric key, the session key is encrypted, and the private asymmetric key is used to decode it. A third method involves encrypting the file with an X.509 certificate, which is used as the symmetric key. A trusted third-party vendor, like VeriSign, issues X.509 certificates.
Case in point: As an illustration of how these encryption techniques may be used in general,

consider the following:
The encrypted XML examples:
<purchaseOrder> \s<Order> \s<Item>book</Item> \s<Id>123-958-74598
</Id> \s<Quantity>12</Quantity> \s</Order> \s<Payment> \s<CardId>123654-8988889
9996874</CardId> \s<CardName>visa</CardName> \s<ValidDate>12-10-2004</ValidDate>\s</Payment> \s</purchaseOrder>
Encrypted version of the preceding XML file:
UTF-8 encoding required for?xml version='1.0'>.
Type="http://www.isi.edu/in-notes/iana/assignments/med ia-types/text/xml" EncryptedData: xmlns=http://www.w3.org/2001/04/xmlenc#
<CipherData> \s<CipherValue>A23B45C56
</CipherValue> \s</CipherData> \s</EncryptedData>
The cypher text, included in CipherData> and CipherValue> tags in the XML file seen above, is encrypted. The contents of the CipherValue> tag, as seen above, contain the encrypted data. An EncryptedData element contains the whole CipherData. The encrypted XML namespace is stored in the EncryptedData element. The other property, xmlns, identifies the namespace used for XML encryption.

Methods of Information Decryption:
The decryption of an encrypted file conforms to a World Wide Web Consortium (W3C) standard. Decryption, in other words, "reverses the encryption process, restoring the original data." This implies that the ciphertext may be deciphered and the original message reconstructed [4]. This is sometimes referred to as "decryption." In most cases, decryption is carried out after encryption, or vice versa. These two techniques are necessary for developing a secure system and meeting the security standards. In this situation, a decryption procedure is applied to the ciphertext under the direction of a decryption key.

D [ciphertext | decrypt key] = E [plaintext| encrypt key] | decrypt key is the formula for decryption.
The equation D [ciphertext | decrypt key] = plaintext (3).
Where "D" stands for the decipherment.
It is the "Decrypt key" that will be used to decrypt the data. [3] Data Signature Methods:

Digital signatures (also known as Dsig or XML-Sig) are used to authenticate the authenticity of documents online. Using this approach ensures the reliability of the data being stored. The file is signed by first calculating its message digest, and only then is the message digest encrypted using the sender's private key. As indicated in the accompanying diagram, the message and digital signature are subsequently sent to the appropriate recipient(s). This section satisfies the prerequisite for putting your signature on specific sections of documents using Transforms, which include pre-processing the document using tools like XPath and so on. You may use it to designate a signature across several documents or portions of documents.

Methods for Evaluating Data:
Validation is essential once Data Signatures have been applied. Decryption, the inverse of encryption, may be used to quickly and simply verify electronic signatures. Decryption may

alternatively be thought of as the receiver's verification of the authenticity and integrity of the data contained inside the encrypted file. A true or false statement is always produced by the file validation process. When a file is validated, the signed information is stripped off at the gateway, revealing the actual file. Verifying the signature requires the sender's public signature key and the same hash function used to create the signature.

The Verification Process of a Digital Signature

As can be seen in the above diagram, the recipient is given both the message digest and the digital signature. The processes used by the receiver to validate the digital signatures are as follows: First, you'll need to locate the file's message digest.

In Step 2, we use the sender's public key to decrypt the digital signatures and get the message digest using the digest algorithm, which is defined in the SignatureMethod element. Checking the signature on the SignedInfo element is done here. In the third step, we'll recompare the digests of the two messages we just examined (in Steps 1 and 2) with the digest of the reference found in the SignedInfo element. Both message digests should be the same if the file's contents haven't been altered. Using a number of different algorithms to encrypt the text, or "plaintext," is step four. "An encryption method executes mathematical operations to make replacements and transformations," says encryptopedia. There are several applications for algorithms in maintaining the security of communications. When using a shared means for communication, safety measures must always be taken. Even though there are a wide variety of algorithms used for encryption, they have some characteristics. Classification of algorithms is possible according on the method and strategy used.

Hash Algorithms (e.g., Message Digest) Key Management Asymmetric Algorithms (e.g., RSA) RSA (Rivest, Shamir, and Adleman) and DSS (Digital Signature Standard) are used for digital signatures, whereas AES (Advanced Encryption Standard) and DES (Data Encryption Standard) are used for encryption. With the help of these algorithms, we can construct the right method-calling function to encrypt, sign, decrypt, and validate the document in question. Rijndael is another name for AES in the cryptography community. Variable block and key lengths characterise this symmetric block cypher. Information may be encrypted using a symmetric block cypher and then decrypted using the same method. Due to its superior sophistication, this algorithm is more difficult to implement than its counterparts. As at the turn of the millennium, AES is still competing. National Institute of Standards and Technology (NIST) released the AES in 2001. It's possible that AES is even quicker than DES, and it's certainly considerably faster than triple-DES. Data is encrypted and decrypted in 128-bit blocks thanks to the Rijndael proposal for AES, which described a cypher in which the block length and key length are separately specced to be 128, 192, or 256 bits. The AES standard restricts the block length to 128 bits and makes use of the same three key size options. Encryption is the primary use for the AES algorithm. This algorithm serves primarily as a means of securing digital information. As was said before, the technique is flexible enough to be employed with keys of three various lengths. Therefore, "AES-128," "AES-192," and "AES-256" are the names given to these variations.

The following are some of the outlined requirements for this:
Parameter, symbol, and function definitions in algorithms; I. Acronym and word definitions;

Second, the algorithm's notation and conventions, such as the bit, byte, and word ordering and numbering schemes.

Mathematical characteristics that aid in comprehending the method
Key length support, keying limits, and new Block/key/round sizes are all examples of implementation problems. Here is how the encrypted version of a document is expected to look after the algorithm has been applied [5]. The contents of the encrypted data are as follows:

The encrypted data's identifier, type, and mime type, and the encryption method and algorithm. When encrypting data using a symmetric approach, it is necessary to indicate that method using the EncryptionMethod element. To do this, it makes use of an Algorithm property that specifies a W3 URL detailing the algorithm used. KeyInfo is used to keep track of details about symmetric keys. Data pertaining to multiple keys may be stored in this component as well. The cypher data is located in the CipherData and CipherValue components, both of which are a part of the EncryptedKey and EncryptedData elements. CipherValue is where the secret information is kept. The CipherValue of the EncryptedData element contains the encrypted data, whereas the EncryptedKey element keeps the encrypted key. To learn more about a certain key, go no farther than the KeyInfo element, which is a child of the EncryptedKey element[6]. The KeyInfo's EncryptionMethod field specifies the asymmetric encryption algorithm used to encrypt the session key. The Algorithm property, when set to a W3C URL, will serve as a suitable replacement. DES: Data Encryption Standard, or DEA for short, is mostly used in computer security. Based on studies conducted by IBM, the Data Encryption Standard (DES) has become the standard private key encryption technique. You may use it to encrypt and decode binary data. Encrypting data changes it to an unreadable form called cypher. Decrypting cypher transforms the data back to its original form called plaintext. Many people are interested in finding a replacement for DES because of its probable susceptibility to a brute-force assault. In 1977, the United States government officially recognised DES as a de facto encryption standard. On March 17, 1975, it was also published in the Federal Register. This method is almost 20 years old and is now considered to be mostly outdated owing to its inadequate key-space. Data Encryption Standard (DES) and Triple Data Encryption Algorithm (TDEA) are two possible cryptographic methods utilised by the United States Federal government[7]. Data security is the responsibility of all organisations. It may be important to safeguard data during transmission or storage in order to protect the privacy and reliability of the information it represents. Algorithms specify in a one-of-a-kind way the sequence of mathematical operations needed to encrypt data and decrypt it again. As part of a comprehensive security programme that also includes training in appropriate information management and computer system/network access restrictions, the Data Encryption Standard is now at your disposal. Using a 56-bit encryption key, DES encrypts 64-bit data blocks through many rounds of transposition and replacement. Notation for DES encryption and decryption is as follows:

Encryption Key: EDES [simple text | key]
Plaintext = DDES [ciphertext | key] + DDES [plaintext | key] + EDES [ciphertext | key]
Where,

The letter "E" represents encryption, whereas the letter "D" represents decryption. An encryption key, or "Key," is a piece of information that may be used to either encrypt or decode data.

Encryption and decryption in DES are performed using the same key. Adding extra steps to the DES encryption process, such as a triple-DES scheme, increases its security. In either hardware or software, DES is not too difficult to implement. Overall, DES has performed better in the face of novel cryptanalytic assaults than many subsequent symmetric cyphers. Therefore, it is integrated into a wide variety of programmes. When compared to traditional cyphers, DES is quite complex. Common implementations use just two steps of encryption and a single stage of decryption, as opposed to the recommended three. The following purpose elaborates on this premise:

DDES [EDES [plain text | key1] | key2] | key1] EDES = encoded text
Decryption, Encryption, and Transcryption; therefore, DESede. So, in this scenario, we just need to worry about using two keys: (key1 and key2). If both keys are the identical, then it operates as a simpler, single-stage DES. Because of this, it is compatible with the older DES encryption method. U.S. government entities may cite this norm as applicable in the following circumstances:

Instance I: Cryptographic protection is mandated by a data security or computer system security officer or management. Neither the National Security Act of 1947, as modified, nor the Atomic Energy Act of 1954, as amended, apply to the information in question.

DSA:

DSS is another name for DSA (Digital Signature Standard). DSA implements an algorithm tailored specifically to the needs of digital signatures. It lacks the key-exchange and encryption capabilities of RSA. However, as this is a public key technique, the signature verification process involves using the public key to decode the signature and then comparing the results to the original message hash calculated using the secret key, as seen in the figure. Cryptography and computer security rely heavily on this. Federal Information Processing Standard FIPS 186, often known as DSS, has been released by the National Institute of Standards and Technology (NIST) [8]. When using the DSA digital signature, two very big integers are used, each of which is represented in the computer as a string of binary digits. In this case, the DSA is relied upon to both create and validate signatures. A digital signature may be created with the help of a private key, while it can be verified with the help of a public key that matches to but is different from the private key.

A symmetric key consists of two halves, the private key and the public key, and is held in a pair by each user. It is presumed that everyone has access to public keys. Not even the two of you will ever see each other's private key [9]. Using a user's public key, any third party may validate the user's signature. Only the owner of the user's private key may generate the user's signature.

Signature Creation and Verification Using DSA and DSS
The hash function is seen in use in the above diagram for both creating and verifying signatures. The DSA takes the message digest as an input and outputs the digital signature. The recipient of the signed data and the digital signature are both sent to the appropriate verifier (often called the message). The recipient's public key is used to verify the authenticity of the signature created by the sender. Verification must also make use of the same hash function[10]. Secure Hash Standard

(SHS), a FIPS-approved standard, details the hashing algorithm. 180"Signatures for both stored and transmitted data may be generated and verified using the same methods. Create the key pair using the DSA (DSS) technique, and set the size of the key to 512 bits. Generate a random pair of keys using the "DSA" instance of the KeyPairGenerator with kpg = KeyPairGenerator.getInstance; (512) kpg.initialize;
This is the code for generating a key pair: KeyPair kp = kpg.generateKeyPair ();

RSA:

"Public key encryption is often based on number theory, as opposed to the transposition and substitution methods used for private key encryption. The field of public-key cryptography relies on this particular algorithm. It is secure because huge numbers are hard to factor. The RSA cypher encrypts data in chunks, making it a block cypher. Due to the ease of encryption and decryption, it is often used for digital signatures and key exchange are sluggish because they take up a lot of time. Ron Rivest, Adi Shamir, and Len Adleman of MIT came up with one of the first solutions to the problem in 1977 and published it the following year [RIVE78]. To our knowledge, this is the first method that can be used for both signing and encrypting. Given long enough keys and modern implementations, the RSA encryption method is thought to be safe for use in an e-commerce system. Here is how the encrypted version of a document is expected to look after the algorithm has been applied.

Conclusion

One must have reliable, secure communications as the foundation for business transactions as secure transmission becomes an integral part of the expanding electronic business infrastructure. As a result, the system's design is finalised and has been refined to perfection. Common security criteria are spelled out in the Security standards, along with the languages and processing rules necessary to fulfil them. This also makes use of preexisting cryptography and security technology, although in an adapted, extendable, and practically useful manner. A wide variety of programmes and libraries provide these security features, which include signature validation and encryption/decryption. More than 30% of the CPU time may be spent on the actual cryptographic procedures performed by the digital signature operations of Sign and Validate. As a result, one may employ hardware cryptographic accelerators for this purpose to get the necessary speed for performing cryptographic operations. Without all four of these steps, the cryptographic process is unfinished and vulnerable. As a result of these norms, the web service may be effectively implemented by using a variety of application programming interfaces (APIs). Web Services, Digital Rights Management, and other developing applications will need widespread adoption of new security standards before businesses can move their operations online. All of the criteria for a secure system—authentication, authorization, secrecy, integrity, signature, and privacy—have been fulfilled. Each and every one of the system's objectives has been met as of the most recent checkpoint. The whole setup serves as a vehicle for providing a concise overview of the standards and the way in which they interact with one another.

REFERENCES

1. Glissa, G., & Meddeb, A. (2019). 6LowPSec: An end-to-end security protocol for 6LoWPAN. *Ad Hoc Networks*, *82*, 100-112.

2. Shin, D., Yun, K., Kim, J., Astillo, P. V., Kim, J. N., & You, I. (2019). A security protocol for route optimization in DMM-based smart home IoT networks. *IEEE Access*, *7*, 142531-142550.
3. Alshehri, M. D., & Hussain, F. K. (2019). A fuzzy security protocol for trust management in the internet of things (Fuzzy-IoT). *Computing*, *101*(7), 791-818.
4. You, I., Kwon, S., Choudhary, G., Sharma, V., & Seo, J. T. (2018). An enhanced LoRaWAN security protocol for privacy preservation in IoT with a case study on a smart factory-enabled parking system. *Sensors*, *18*(6), 1888.
5. Wang, S., Zhu, S., & Zhang, Y. (2018, June). Blockchain-based mutual authentication security protocol for distributed RFID systems. In *2018 IEEE Symposium on Computers and Communications (ISCC)* (pp. 00074-00077). IEEE.
6. Mezrag, F., Bitam, S., & Mellouk, A. (2019, September). IDSP: A new identity-based security protocol for cluster-based wireless sensor networks. In *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)* (pp. 1-6). IEEE.
7. Abdullah, K. M., Houssein, E. H., & Zayed, H. H. (2018, April). New security protocol using hybrid cryptography algorithm for WSN. In *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)* (pp. 1-6). IEEE.
8. Rath, M., & Pattanayak, B. K. (2019). Security protocol with IDS framework using mobile agent in robotic MANET. *International Journal of Information Security and Privacy (IJISP)*, *13*(1), 46-58.
9. Langer, M., Teichel, K., Sibold, D., & Bermbach, R. (2018, April). Time synchronization performance using the network time security protocol. In *2018 European Frequency and Time Forum (EFTF)* (pp. 138-144). IEEE.
10. Langer, M., Teichel, K., Sibold, D., & Bermbach, R. (2018, April). Time synchronization performance using the network time security protocol. In *2018 European Frequency and Time Forum (EFTF)* (pp. 138-144). IEEE.