Homework #3

Oregon State University

ECE 478 Network Security

Spring 2016

Student:
Sam Quinn
Quinnsa@Oregonstate.edu

Professor:
Dr. Attila A Yavuz
Attila.Yavuz@oregonstate.edu

May 26, 2016

Oregon State
UNIVERSITY

College of Engineering

Contents

1 Impact of the PRGN Quality over the Security of DLP-based Primitives

[30] Consider the following attack:
The server computes a Schnorr signature under private key y (here we follow the notation of ETA slide/paper) periodically such that for every message M to be signed, it adds current time stamp ts_i and involve randomness r_i as $H(M||ts\_i||r\_i)$ and follows the Schnorr signature algorithm as required. Here, r_i is the output of a PRNG, which derives randomness from certain Operating System (OS) parameters.
Assume that an attacker managed infiltrating a virus into the OS of the server, which is capable of resetting OS parameters that PRGN relies on, meaning the seed of PRNG is set to its initial values.
Show that by observing small-constant number of signatures, the attacker can totally break the Schnorr signature scheme under these circumstances (remark that it does not mean the Schnorr signature scheme is flawed, it is about how it is used with certain PRNGs).
You must show how the attack works step by step, by illustrating algebraic recovery step, and explain how ts_i and r_i play a role in this attack.
To sign a message m the signer must:
Let $k = 0 < k < q$
Let $x = 0 < x < q$
Let $r = g^k$
Let $e = H(m\|r)$
Let $s = (k - xe)\ mod\ q$
$\sigma = (s, e)$

If an adversary has compramized the system in which they are able to deterministically invoke a value via a PRGN then the adversary could forge any message with the following. Because the Schorr algorithm uses the random varible r_i as a source of entropy with the hashing mechanism if the adversary is able to produce at least 2 messages with the same r_i they would able to recover the private key. If the private key is derived the adversary can forge any message they please.

Let two signatures be (s, e) and (s', e') for two messages $m \neq m'$, but the random number $r = r'$ because $k = k'$.

$$s \equiv k - xe\ (mod\ q)$$

and

$$s' \equiv k - xe'\ (mod\ q)$$

Since the value of s is the same in both signatures, we can conclude:

$$s + xe \equiv s' + xe' \ (mod\ q)$$

which with some more simple math can extract the private key x with:

$$s - s' \equiv x(e' - e) \ (mod\ q)$$
$$x \equiv (s - s')(e' - e)^{-1} \ (mod\ q)$$

Now that the adversary has extracted the secret key x they may forge any signature they would like, totally breaking the Schnorr Signature scheme.

2 Basic security models for authentication primitives

- **(3) Explain EU-CMA experiment for a MAC scheme.**
  Existential Unforgeable Under Chosen Message Attacks defines a MAC scheme that should prevent any forged tag created by the adversary from being verified.

  1. This experiment creates a key k with the keygen given 1^n.
  2. Then the adversary A is given the input 1^n and access to the Mac_k oracle.
  3. The adversary outputs $(m,\ t)$. Each query that A produces is denoted as Q. The adversary succeeds if and only if $Vrfy_k(m, t) = 1$ and $m \notin Q$.

  If the previous is satisfied then the adversary wins. The probability of this happening is:

  $$Pr[Mac - forge_{A,\pi}(n) = 1] \leq negl(n)$$

  [1]

- **(4) Explain EU-CMA experiment for a digital signature scheme.**

  1. Obtain keys (pk, sk) from the keygen with 1^n.
  2. The adversary A is given only the pk and access to the $Sign_{sk}$ oracle. Each query that the adversary makes is denoted Q.
  3. The adversary succeeds if and only if $Vrfy_{pk}(m, \sigma) = 1$ and $m \notin Q$

  The adversary can achieve the steps listed above then the adversary wins. The probability that the adversary can execute such an attack is:

  $$Pr[Sig - forge_{A,\pi}(n) = 1] \leq negl(n)$$

  [1]

- **(3) Why adaptability is important, and what are non-triviality and validity conditions?**
  With these security games the challenger must be able to protect against adversaries that can adapt to different scenarios. It is important for each scheme to be non-trivial in respects to the probability that the advantage the adversary has.

Please describe the formal experiments with cryptographic games, providing the related probability equation, and then give a brief explanation what these probabilities means. You may refer Mihir Bellares Lecture Notes as a source. Also, the book of Prof. Dr. Jonathan Katz referred in the Syllabus is a good resource for this question.

All signature schemes require a *Keygen*, *Sign*, and a *Verify* function. A cryptographic game that would be able to test the security of a signature scheme is defined as follows.
Setup Let a challenger run *Keygen* and publicly distribute the public key PK to the adversary. The secret key SK will be kept confidential to only the challenger.
Signature Queries The adversary will send messages $m_1, \ldots, m_q$ to the challenger to which they will sign by calling *sign* $m_i \rightarrow \sigma_i$ and send σ back to the adversary. The queries may or may not depend on previous signed messages $m_1, \ldots, m_{i-1}$
Result If the adversary is able to create a pair (m, σ) that has a valid signature of m with the use of *Verify* in the case that m is not included in the original m_i, then the challenger loses and the adversary wins.

3 Synergy between Elgamal and DH

[20] Do you see any synergy between Elgamal encryption scheme and DH key exchange

Elgamal encryption scheme is very similar to the Diffie Hellman key agreement, the only difference is that instead of agreeing on a shared key the data is encrypted. A message that is encrypted under Elgamal must also be $M \in \mathbb{Z}_p$, where $\mathbb{Z}_p$ is the cyclic group. Because the message M needs to be within the cyclic group of the parameters it is considered to have high overhead as an encryption scheme.

Diffie Hellman Key Agreement:

Alice	Transfer	*Bob*
$a \leftarrow \mathbb{Z}_n$		$b \leftarrow \mathbb{Z}_n$
$A = g^a$	$A \rightarrow$	
	$\leftarrow B$	$B = g^b$
$K = B^a$		$K = A^b$

The Diffie Hellman Key Agreement is interactive between the two parties, while the Elgamal encryption scheme the party that is encrypting has an unequal share of computation.

Elgamal Encryption:

Keygen:	ENC(A,M)	DEC(a,(B,C)):
$a \leftarrow \mathbb{Z}_n$	$b \leftarrow \mathbb{Z}_n$	$K = B^a$
$pk = A = g^a$	$B = g^b$	$M = C * K^{-1}$
$sk = a$	$K = A^b$	Return M
	$C = K * M$	
	Return (B, C)	

From the two schemes demonstrated above we can conclude that Elgamal and Diffie Hellman share these same variables:

Elgamal	Diffie Hellman
sk	a or b
pk	A or B

4 RSA digital signatures and their properties

[20] Aggregate signatures enable O(L) protocol? Explain how one implicitly uses another to achieve its objective. signatures to be compressed into a single, compact (i.e., constant-size) verifiable signature.

- **In RSA, it is easy to aggregate signatures computed under the same private key. What is the name of this scheme, and how this aggregation is performed?**
  This RSA scheme is called condensed RSA or C-RSA. With t different messages $m_1, \ldots, m_t$ and their signatures $\sigma_1, \ldots, \sigma_t$ created by the same signer. The Condensed-RSA signature is computed as follows:

$$\sigma_{1,t} = \prod_{i=1}^{t} \sigma_i \ (mod\ n)$$

  [2]

- **In cloud computing environment, especially in outsourced databases, several tuples belonging to different users are stored together. Therefore, it is desirable to aggregate signatures computed with different keys belonging different users. Is it possible to use the RSA aggregation strategy discussed above for this purpose? If it is possible, show how it can be done. If it is not possible, explain the reason in detail.**
  To preserve authentication amunst signatures the signatures must not use multiplication like in C-RSA. When there are multiple different signatures due to RSA's multiplicitavly homomorphic property to aggregate multiple different signatures addition is used. This scheme is called BGLS and is aggergated as described below:

$$e(\sigma_{1,t}, g_2) = \sum_{i=1}^{t} e(H_i, v_i)$$

[2]

Hint: The paper Practical immutable signature bouquets (PISB) for authentication and integrity in outsourced databases might give you clues for both questions.

5 PKI Concepts

- **(4) What are the core components of a PKI? Briefly describe each component.**

  A Certificate Authority (CA)
  Digitally sign and publish public keys for users.

  A Registration Authority
  Verifies identities of users requesting their digital certificates to be stored in the CA.

  A Central Directory
  A secure storage location to store and index keys.

  A Certificate Management System
  Manages access and delivery of certificates.

  A Certificate Policy
  A published document explaining the architecture of the PKI and the roles.

- **(3) Discuss the trustworthiness of root certificates provided by browsers.**
  Browsers come with a preinstalled list of root certificate authorities. The trustworthiness is solely based on the users trust of the given browser. All certificates below the root certificate inherits the trust of the root. The certificates for browsers verify identities that use SSL connections.

- **(3) What is the purpose of the X.509 standard and what is a certificate chain? How is an X.509 certificate revoked?**
  The X.509 is a standard for PKI's to manage digital certificates. A certificate chain is a series of certificates that form a path to a trusted root. Each certificate will be signed by the secret key of the next certificate in the chain leading up to the last certificate. The last certificate is often self-signed. To revoke a certificate the certificate unique serial number is added to a black list. When a X.509 certificate is revoked the client will notice that the certificate is in the blacklist and should not be trusted, however the blacklists are often not maintained as well as they should.

References

[1] J. Katz and Y. Lindell, *Introduction to Modern Cryptography, Second Edition.* Chapman & Hall/CRC Cryptography and Network Security Series, Taylor & Francis, 2014.

[2] E. Mykletun, M. Narasimha, and G. Tsudik, "Signature bouquets: Immutability for aggregated/condensed signatures," in *Computer Security–ESORICS 2004*, pp. 160–176, Springer, 2004.