Lesson 9
Digital Signatures

UNIT 1
Public key Digital signatures

Digital signatures

Encryption scheme: solves the problem of secure communication over insecure network

Digital signature: solves a different problem, analogous to the purpose of a pen-and-ink signature on a physical document.

Digital signatures

THE PROBLEM

Samantha has a digital document *D* (ex. computer file) and she wants to create some additional piece of information D^{sign} that can be used to prove conclusively that Samantha herself approves of the document.

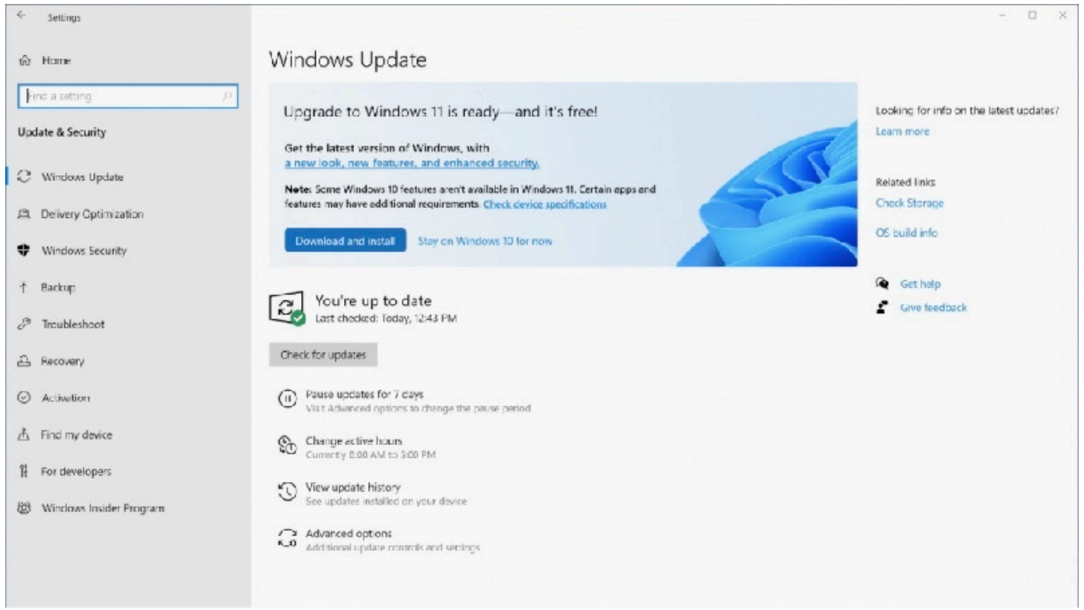
Samantha's digital signature D^{sign} analogous to her actual signature on an ordinary piece of document.

Victor, who receives the document and the signature, needs to verify it.

Importance of Digital signatures

Digital signatures are at least as important as public key cryptosystems for the conduct of business in a digital age.

Your computer receives program and system upgrades over the Internet.



How can your computer tell that an upgrade comes from a legitimate source, in this case the company that wrote the program in the first place?

Importance of Digital signatures

The original program comes equipped with the company's public verification key.

The company uses its private signing key to sign the upgrade and sends your computer both the new program and the signature.

Your computer can use the public key to verify the signature,

thereby verifying that the program comes from a trusted source, before installing it on your system.

Recall: Asymmetric cipher

DEFINITION (asymmetric cipher)

An **asymmetric cipher** is a 5-uple $(\mathcal{K}, \mathcal{M}, \mathcal{C}, e, d)$ where

$\mathcal{K} \subset \mathcal{K}_{\text{priv}} \times \mathcal{K}_{\text{pub}}$: space of keys

$\mathcal{K}_{\text{priv}}$: private keys

$\mathcal{K}_{\text{pub}}$: public keys

$\mathcal{M}$: space of messages

$\mathcal{C}$: space of ciphertexts

encryption function

$$e : \mathcal{K}_{\text{pub}} \times \mathcal{M} \to \mathcal{C}$$

decryption function

$$d : \mathcal{K}_{\text{priv}} \times \mathcal{C} \to \mathcal{M}$$

$$\forall (k_{\text{priv}}, k_{\text{pub}}) \in \mathcal{K} \qquad \forall m \in \mathcal{M}$$

$$\boxed{d(k_{\text{priv}}, e(k_{\text{pub}}, m)) = m}$$

The aim: transmit secretly *m*

Public key Digital signatures

An asymmetric cipher is a 5-uple $(\mathcal{K}, \mathcal{M}, \mathcal{C}, e, d)$ where

$\mathcal{K} \subset \mathcal{K}_{\text{priv}} \times \mathcal{K}_{\text{pub}}$: space of keys

$\mathcal{K}_{\text{priv}}$: private keys

$\mathcal{K}_{\text{pub}}$: public keys

$\mathcal{M}$: space of messages

$\mathcal{C}$: space of ciphertexts

encryption function $e : \mathcal{K}_{\text{pub}} \times \mathcal{M} \to \mathcal{C}$

decryption function $d : \mathcal{K}_{\text{priv}} \times \mathcal{C} \to \mathcal{M}$

$\forall (k_{\text{priv}}, k_{\text{pub}}) \in \mathcal{K} \qquad \forall m \in \mathcal{M}$

$$d(k_{\text{priv}}, e(k_{\text{pub}}, m)) = m$$

DEFINITION (Digital signatures)

A digital signature scheme is a 5-uple $(\mathcal{K}, \mathcal{D}, \mathcal{S}, \sigma, \nu)$ where

$\mathcal{K} \subset \mathcal{K}_{\text{priv}} \times \mathcal{K}_{\text{pub}}$: space of keys

$\mathcal{D}$: documents $\quad \mathcal{S}$: signatures

$\mathcal{K}_{\text{priv}}$: private signing keys

$\mathcal{K}_{\text{pub}}$: public verification keys

signing function

$$\sigma : \mathcal{K}_{\text{priv}} \times \mathcal{D} \longrightarrow \mathcal{S}$$

$$(k_{\text{priv}}, D) \longmapsto D^{\text{sign}} := \sigma(k_{\text{priv}}, D)$$

verification function

$$\nu : \mathcal{K}_{\text{pub}} \times (\mathcal{D} \times \mathcal{S}) \longrightarrow \{\text{True, False}\}$$

$$(k_{\text{pub}}, (D, S)) \longmapsto \nu(k_{\text{pub}}, (D, S))$$

$$\forall (k_{\text{priv}}, k_{\text{pub}}) \in \mathcal{K} \qquad \forall D \in \mathcal{D}$$

$$\nu(k_{\text{pub}}, (D, \sigma(k_{\text{priv}}, D))) = \text{True}$$

True if D signed by k_{priv} is D^{sign}

The signed document is $(D, \sigma(k_{\text{priv}}, D))$. The signature on D by k_{priv} is $\sigma(k_{\text{priv}}, D)$.

The aim: verifying the authenticity of digital messages or documents

Public key Digital signatures

An asymmetric cipher is a 5-uple $(\mathcal{K}, \mathcal{M}, \mathcal{C}, e, d)$ where

$\mathcal{K} \subset \mathcal{K}_{\text{priv}} \times \mathcal{K}_{\text{pub}}$: space of keys

$\mathcal{K}_{\text{priv}}$: private keys

$\mathcal{K}_{\text{pub}}$: public keys

$\mathcal{M}$: space of messages

$\mathcal{C}$: space of ciphertexts

encryption function

$$e : \mathcal{K}_{\text{pub}} \times \mathcal{M} \to \mathcal{C}$$

decryption function

$$d : \mathcal{K}_{\text{priv}} \times \mathcal{C} \to \mathcal{M}$$

$$\forall (k_{\text{priv}}, k_{\text{pub}}) \in \mathcal{K} \qquad \forall m \in \mathcal{M}$$

$$d(k_{\text{priv}}, e(k_{\text{pub}}, m)) = m$$

A digital signature scheme is a 5-uple $(\mathcal{K}, \mathcal{D}, \mathcal{S}, \sigma, \nu)$ where

$\mathcal{K} \subset \mathcal{K}_{\text{priv}} \times \mathcal{K}_{\text{pub}}$: space of keys

$\mathcal{K}_{\text{priv}}$: private signing keys

$\mathcal{K}_{\text{pub}}$: public verification keys

$\mathcal{D}$: documents

$\mathcal{S}$: signed documents

signing function

$$\sigma : \mathcal{K}_{\text{priv}} \times \mathcal{D} \longrightarrow \mathcal{S}$$

$$(k_{\text{priv}}, D) \longmapsto D^{\text{sign}} := \sigma(k_{\text{priv}}, D)$$

verification function

$$\nu : \mathcal{K}_{\text{pub}} \times (\mathcal{D} \times \mathcal{S}) \longrightarrow \{\text{True}, \text{False}\}$$

$$(k_{\text{pub}}, (D, S)) \longmapsto \nu(k_{\text{pub}}, (D, S))$$

$$\forall (k_{\text{priv}}, k_{\text{pub}}) \in \mathcal{K} \qquad \forall D \in \mathcal{D}$$

$$\nu(k_{\text{pub}}, (D, \sigma(k_{\text{priv}}, D))) = \text{True}$$

Public key Digital signatures

A digital signature scheme is a 5-uple $(\mathcal{K}, \mathcal{D}, \mathcal{S}, \sigma, \nu)$ where

$\mathcal{K} \subset \mathcal{K}_{\text{priv}} \times \mathcal{K}_{\text{pub}}$: space of keys
$\mathcal{K}_{\text{priv}}$: private signing keys
$\mathcal{K}_{\text{pub}}$: public verification keys

$\mathcal{D}$: documents
$\mathcal{S}$: signed documents

signing function

$$\sigma : \mathcal{K}_{\text{priv}} \times \mathcal{D} \longrightarrow \mathcal{S}$$
$$(k_{\text{priv}}, D) \longmapsto D^{\text{sign}} := \sigma(k_{\text{priv}}, D)$$

verification function

$$\nu : \mathcal{K}_{\text{pub}} \times (\mathcal{D} \times \mathcal{S}) \longrightarrow \{\text{True, False}\}$$
$$(k_{\text{pub}}, (D, S)) \longmapsto \nu(k_{\text{pub}}, (D, S))$$

$$\forall (k_{\text{priv}}, k_{\text{pub}}) \in \mathcal{K} \qquad \forall D \in \mathcal{D} \qquad \nu(k_{\text{pub}}, (D, \sigma(k_{\text{priv}}, D))) = \text{True}$$

REMARK

It is easy to produce useless digital schemes.

If $k_{\text{pub}} = k_{\text{priv}}$ any attacker can produce the same signature.

EXAMPLE

Name	Size	Kind	Date Added
SignatureMariconda.jpg	49 KB	JPEG image	Today at 09:32

$k_{\text{priv}} = k_{\text{pub}} =$SignatureMariconda.jpg: signature file

$\nu(\text{SignatureMariconda.jpg}, (D, S)) =$ "True" if S is obtained by pasting SignatureMariconda.jpg on D.

Secure Public key Digital signatures

A digital signature scheme is a 5-uple $(\mathcal{K}, \mathcal{D}, \mathcal{S}, \sigma, \nu)$ where

$\mathcal{K} \subset \mathcal{K}_{\text{priv}} \times \mathcal{K}_{\text{pub}}$: space of keys
$\mathcal{K}_{\text{priv}}$: private signing keys
$\mathcal{K}_{\text{pub}}$: public verification keys

$\mathcal{D}$: documents
$\mathcal{S}$: signed documents

signing function

$$\sigma : \mathcal{K}_{\text{priv}} \times \mathcal{D} \longrightarrow \mathcal{S}$$
$$(k_{\text{priv}}, D) \longmapsto D^{\text{sign}} := \sigma(k_{\text{priv}}, D)$$

verification function

$$\nu : \mathcal{K}_{\text{pub}} \times (\mathcal{D} \times \mathcal{S}) \longrightarrow \{\text{True, False}\}$$
$$(k_{\text{pub}}, (D, S)) \longmapsto \nu(k_{\text{pub}}, (D, S))$$

$$\forall (k_{\text{priv}}, k_{\text{pub}}) \in \mathcal{K} \qquad \forall D \in \mathcal{D} \qquad \nu(k_{\text{pub}}, (D, \sigma(k_{\text{priv}}, D))) = \text{True}$$

SECURE DIGITAL SIGNATURE SCHEME

1. Given k_{pub}, an attacker cannot feasibly determine k_{priv}, nor can she determine any other private key that produces the same signatures as k_{priv}.

2. Given k_{pub} and a list of signed documents $D_1, \ldots, D_n$ together with their signatures $D_1^{\text{sign}}, \ldots, D_n^{\text{sign}}$, an attacker cannot feasibly determine a valid signature on any document D that is not in the list $D_1, \ldots, D_n$.

UNIT 2
RSA type Digital Signature

Recall: RSA Public key Cryptosystem

Bob wants to send a message to Alice in such a way that just Alice is able to read.

Rivest-Shamir-Adleman Public key cryptosystem (1977)

SETUP

1. **Alice chooses two large distinct primes p, q and computes $N = pq$,**

   $\phi(N) = (p-1)(q-1)$. SECRET

2. **Alice chooses $e \in (\mathbb{Z}/\phi(N)\mathbb{Z})^*$** ($\gcd(e, \phi(N)) = 1$) **and computes $d := e^{-1}$.** SECRET

   decryption exponent

3. **Alice makes N (modulus) and e (encryption key) public.** PUBLIC

ENCRYPTION

4. **Bob sends $c = m^e$ in $\mathbb{Z}/N\mathbb{Z}$ to Alice.** Might be PUBLIC

DECRYPTION

5. **Alice computes $m = c^d$ in $\mathbb{Z}/N\mathbb{Z}$.**

RSA type Digital signature

Samantha wants to send to Victor a document together with her digital signature.

Rivest-Shamir-Adleman Public key Digital Signature (1977)

SETUP

1. **Samantha chooses two large distinct primes p, q and computes $N = pq$,**
   $\phi(N) = (p-1)(q-1)$. SECRET
2. **Samantha chooses $e \in (\mathbb{Z}/\phi(N)\mathbb{Z})^*$ and computes $d := e^{-1}$.** SECRET
   $\gcd(e, \phi(N)) = 1$
   private signing key
3. **Samantha makes N (modulus) and e (public verification exponent) public.** PUBLIC

SIGNING

4. **Samantha signs document D by computing $S = D^d[N]$: she sends (D, S).** Might be PUBLIC

VERIFICATION

5. **Victor computes $S^e[N]$.** $S^e \begin{cases} = D: & \text{True} \\ \neq D: & \text{False.} \end{cases}$

CORRECTNESS **The process works because $S^e \equiv D^{de} \equiv D[N]$: indeed $ed \equiv 1[\phi(N)]$.**

RSA type Digital signature

SETUP

1. Samantha chooses two large distinct primes p, q and computes $N = pq$, $\phi(N) = (p-1)(q-1)$. SECRET
2. Samantha chooses $e \in (\mathbb{Z}/\phi(N)\mathbb{Z})^*$ ($\gcd(e, \phi(N)) = 1$) and computes $d := e^{-1}$. SECRET — d: private signing key
3. Samantha makes N (modulus) and e (public verification exponent) public. PUBLIC

SIGNING

4. Samantha signs document D by computing $S = D^d[N]$: she sends (D, S). Might be PUBLIC

VERIFICATION

5. Victor computes $S^e[N]$. $S^e \begin{cases} = D: \text{ True} \\ \neq D: \text{ False.} \end{cases}$

REMARK

The process Samantha follows closely *resembles that of the RSA cipher*. Essentially, she performs the same steps as she would in decrypting a message from her partner. However, the difference is that these steps are undertaken *prior* to sending the document, rather than after receiving the cipher, as is the case with the RSA cipher.

It works because the encryption and decryption function commute:

$$(m^e)^d = (m^d)^e$$

RSA type Digital signature

SETUP

1. Samantha chooses two large distinct primes p, q and computes $N = pq$, $\phi(N) = (p-1)(q-1)$. SECRET
2. Samantha chooses $e \in (\mathbb{Z}/\phi(N)\mathbb{Z})^*$ and computes $d := e^{-1}$. SECRET (private signing key)
   $\gcd(e, \phi(N)) = 1$
3. Samantha makes N (modulus) and e (public verification exponent) public. PUBLIC

SIGNING

4. Samantha signs document D by computing $S = D^d[N]$: she sends (D, S). Might be PUBLIC

VERIFICATION

5. Victor computes $S^e[N]$. $S^e \begin{cases} = D: \text{ True} \\ \neq D: \text{ False.} \end{cases}$

EXAMPLE SETUP

1. Samantha chooses two secret primes $p = 1223$ and $q = 1987$ and computes

$$N = pq = 1223 \cdot 1987 = 2430101,$$

2. Samantha chooses $e = 948047 \in \mathbb{Z}/\phi(N)\mathbb{Z}$ and computes $d = e^{-1}$ in $\mathbb{Z}/\phi(N)\mathbb{Z}$:

3. Samantha makes N, e public.

SIGNING 

4. Samantha signs a document $1 < D = 1070777 < N$:

RSA VERIFICATION

RSA type Digital signature: Eve's point of view

SETUP

1. Samantha chooses two large distinct primes p, q and computes $N = pq$, $\phi(N) = (p-1)(q-1)$. SECRET
2. Samantha chooses $e \in (\mathbb{Z}/\phi(N)\mathbb{Z})^*$ and computes $d := e^{-1}$. SECRET (private signing key)
   $\gcd(e, \phi(N)) = 1$
3. Samantha makes N (modulus) and e (public verification exponent) public. PUBLIC

SIGNING

4. Samantha signs document D by computing $S = D^d[N]$: she sends (D, S). Might be PUBLIC

VERIFICATION

5. Victor computes $S^e[N]$. $S^e \begin{cases} = D: \text{ True} \\ \neq D: \text{ False.} \end{cases}$

REMARK

If Eve can factor N, then she can solve *ed*=1 [*N*], for Samantha's secret signing key *d*. However, just as with RSA encryption, the hard problem underlying RSA digital signatures is not directly the problem of factorization.

In order to forge a signature on a document *D*, Eve needs to find a *e*-th root of *D* modulo *N*. This is *identical to the hard problem underlying RSA decryption*, in which the plaintext is the *e*-th root of the ciphertext.

UNIT 3
Elgamal type Digital Signature

Recall: Elgamal Public key Cryptosystem

Bob wants to send a message to Alice in such a way that just Alice is able to read.

Elgamal Public key cryptosystem (1984)

SETUP

1. **Alice chooses a large prime p and $g \in \mathbb{F}_p^*$ of large order.** PUBLIC
2. **Alice chooses a secret number a** PRIVATE KEY
3. **Alice publishes $A = g^a$ in $\mathbb{F}_p$.** PUBLIC KEY

ENCRYPTION: ^k

PRIVATE, used once to randomize the encryption process

4. **Bob wants to send a message $m \in \mathbb{F}_p$: he chooses $1 < k < p-1$ randomly and sends $(c_1 = g^k, c_2 = mA^k) \in \mathbb{F}_p \times \mathbb{F}_p$.** Might be PUBLIC

DECRYPTION:^a

5. **Alice computes $m = (c_1^a)^{-1} \cdot c_2$.**

It works because $(c_1^a)^{-1} \cdot c_2 = g^{-ka} \cdot mA^k = g^{-ka} \cdot m(g^a)^k = m$.

Towards a Elgamal Digital Signature scheme

Elgamal Public key cryptosystem (1984)

SETUP

1. Alice chooses a large prime p and $g \in \mathbb{F}_p^*$ of large order. PUBLIC
2. Alice chooses a secret number a PRIVATE KEY
3. Alice publishes $A = g^a$ in $\mathbb{F}_p$. PUBLIC KEY

ENCRYPTION: ^k

PRIVATE, used once to randomize the encryption process

4. Bob wants to send a message $m \in \mathbb{F}_p$: he chooses $1 < k < p-1$ randomly and sends $(c_1 = g^k, c_2 = mA^k) \in \mathbb{F}_p \times \mathbb{F}_p$. Might be PUBLIC

DECRYPTION:^a

5. Alice computes $m = (c_1^a)^{-1} \cdot c_2$.

Think at RSA: Samantha does the same job as in the cipher, prior to sending the document.

Here she should compute $(c_1^a)^{-1} \cdot c_2$, where $c_1 = g^k$, $c_2 = DA^k$.

But...k...? wasn't it chosen by Victor? And what Victor should do?

Problem: encryption and decryption functions do not commute: once k is chosen,

$$e_{A,g}(m) = (g^k, mA^k);\ d_a(c_1, c_2) = (c_1^a)^{-1} \cdot c_2. \qquad d_a(e_{A,g}(m)) = m,$$

$e_{A,g} \circ d_a$ acts on pairs (c_1, c_2): $e_{A,g} \circ d_a \neq d_a \circ e_{A,g}$.

The Elgamal Digital Signature scheme

DO NOT REMEMBER THE RULE

Samantha wants to send to Victor a document together with her digital signature.

Elgamal Digital Signature Scheme

SETUP

1. Samantha or a trusted party chooses a large prime p and a primitive root $g \in \mathbb{F}_p^*$. PUBLIC
2. Samantha chooses a private signing key $1 < a < p-1$. She computes $A = g^a[p]$. PRIVATE KEY
3. Samantha publishes the public verification key $A = g^a$ in $\mathbb{F}_p$. PUBLIC KEY

SIGNING with k

PRIVATE, used once to randomize the encryption process

4. Samantha chooses a document $D[p]$, and a random element k: $\gcd(k, p-1) = 1$.
   She signs $S_1 = g^k[p], S_2 = (D - aS_1)k^{-1}[p-1]$. She sends $(D, (S_1, S_2))$.

VERIFICATION

5. Victor computes $A^{S_1} S_1^{S_2}[p]$ and g^D.

$$A^{S_1} S_1^{S_2} \begin{cases} \equiv g^D : \text{ True} \\ \not\equiv g^D : \text{ False.} \end{cases}$$

CORRECTNESS It works because $A^{S_1} S_1^{S_2} = g^{aS_1} g^{kS_2} = g^{aS_1} g^{D-aS_1} = g^D$.

The Elgamal Digital Signature scheme

SETUP

1. Samantha or a trusted party chooses a large prime p and a primitive root $g \in \mathbb{F}_p^*$. PUBLIC
2. Samantha chooses a private signing key $1 < a < p-1$. She computes $A = g^a[p]$. PRIVATE KEY
3. Samantha publishes the public verification key $A = g^a$ in $\mathbb{F}_p$. PUBLIC KEY

SIGNING with k  PRIVATE, used once to randomize the encryption process

4. Samantha chooses a document $D[p]$, and a random element k: $\gcd(k, p-1) = 1$. She signs $S_1 = g^k[p]$, $S_2 = (D - aS_1)k^{-1}[p-1]$. She sends $(D, (S_1, S_2))$.

VERIFICATION

5. Victor computes $A^{S_1}S_1^{S_2}[p]$ and g^D. $A^{S_1}S_1^{S_2} \begin{cases} \equiv g^D: \text{ True} \\ \not\equiv g^D: \text{ False.} \end{cases}$

EXAMPLE SETUP

1. Samantha chooses the prime $p = 21739$ and primitive root $g = 7$.
2. Samantha chooses the secret key $a = 15140$ and computes her public verification key $A \equiv g^a[p] \equiv 7^{15140}[21739] \equiv 17702[21739]$.
3. Samantha publishes $A \equiv 17702[21739]$.

SIGNING

VERIFICATION

The Elgamal Digital Signature scheme: Eve's point

SETUP

1. Samantha or a trusted party chooses a large prime p and a primitive root $g \in \mathbb{F}_p^*$. PUBLIC
2. Samantha chooses a private signing key $1 < a < p-1$. She computes $A = g^a[p]$. PRIVATE KEY
3. Samantha publishes the public verification key $A = g^a$ in $\mathbb{F}_p$. PUBLIC KEY

SIGNING with k  PRIVATE, used once to randomize the encryption process

4. Samantha chooses a document $D[p]$, and a random element k: $\gcd(k, p-1) = 1$.
   She signs $S_1 = g^k[p]$, $S_2 = (D - aS_1)k^{-1}[p-1]$. She sends $(D, (S_1, S_2))$.

VERIFICATION

5. Victor computes $A^{S_1} S_1^{S_2}[p]$ and g^D. $A^{S_1} S_1^{S_2} \begin{cases} \equiv g^D: \text{ True} \\ \not\equiv g^D: \text{ False.} \end{cases}$

REMARK

If Eve knows how to solve the DLP she can solve

$$g^a \equiv A[p]$$

for Samantha's private signing key *a*,
thus she can forge Samantha's signature (she may take an arbitrary value *k*).

Another way: Given A, g^D, Eve must find integers x, y satisfying: $A^x x^y = g^D[p]$.

This is equivalent to $x \log_g(A) + y \log_g(x) \equiv D[p-1]$.

One way to solve: knowing $\log_g(A)$, choose arbitrary x, whence y.

DLP again!

Are there other ways to solve that bizarre equation?

L9 - U3

END of
Lesson 9
Digital Signatures