Communicating Through Subliminal-Free Signatures

George Teșeleanu

Advanced Technologies Institute

Simion Stoilow Institute of Mathematics

November, 2021

Outline

Prisoners' problem

- *Alice* (sender) and *Bob* (receiver) are incarcerated.
- They want to communicate confidentially and undetected by their guard *Walter*.
- *Walter* imposes to read all their communication.
- Subliminal channels are a possible solution to the prisoners' problem.
- Achieves information transfer by modifying the original specifications of cryptographic primitives.
- An example: modify the way random numbers are generated.

Description

Public Parameters' Generation

- Select a prime number $q \geq 2^{\kappa}$.
- Select a prime number $p \geq 2^{\lambda}$ such that $q|p-1$.
- Choose an element $g \in \mathbb{Z}_p$ of order q.
- Choose a hash function $h : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$.
- Output the public parameters $pp = (p, q, g, h)$.

Description

Signer's Key Generation

- Choose $x \xleftarrow{\$} \mathbb{Z}_q^*$.
- Compute $y \leftarrow g^x \bmod p$.
- Output the public key $pk = y$ and the secret key $sk = x$.

Description

Alice		*Walter*
	step ①	
Choose $k' \xleftarrow{\$} \mathbb{Z}^*_{p-1}$. Compute $t \leftarrow g^{k'} \bmod p$.		
	$\xrightarrow{t}$	
	step ②	
		Choose $k'' \xleftarrow{\$} \mathbb{Z}^*_{p-1}$.
	$\xleftarrow{k''}$	
	step ③	
Compute the following: $k \leftarrow k'k'' \bmod p-1$, $r \leftarrow (g^k \bmod p) \bmod q$, $s \leftarrow k^{-1}[h(m)+xr] \bmod q$.		
	$\xrightarrow{(m,r,s)}$	
	step ④	
		If $r \neq (t^{k''} \bmod p) \bmod q$, `abort`.

Description

Verification

- Compute $u_1 \leftarrow h(m)s^{-1} \bmod q$ and $u_2 \leftarrow rs^{-1} \bmod q$.
- Compute $v \leftarrow (g^{u_1} y^{u_2} \bmod p) \bmod q$.
- Output `true` if and only if $v = r$. Otherwise, output `false`.

Introduction

- To communicate ω to *Bob*, *Alice* must stop the protocol if certain conditions are not achieved.
- If the protocol is stopped too often by *Alice*, *Walter* might become suspicious and cut off any communication between the prisoners.
- *Alice* can only send a few bits of data to *Bob* through this channel.

Description

Alice *Walter*

step ③

Compute the following:
$k \leftarrow k'k'' \bmod p-1$,
$r \leftarrow (g^k \bmod p) \bmod q$,
$s \leftarrow k^{-1}[h(m)+xr] \bmod q$.
If $\omega \not\equiv r \bmod 2$ `abort`.

$\xrightarrow{(m,r,s)}$

Description

Extract

- To extract the embedded message ω compute $\omega \leftarrow r \bmod 2$.

Introduction

- Compared to fail-stop channels, cuckoo's channels are used by a dishonest *Walter* to convey information to a third party.
- Just like a cuckoo that lays his eggs in the nests of unsuspecting birds, *Walter* inserts his message into *Alice*'s signature without her suspecting anything.

Description

Alice *Walter*

step ④

Choose $k'' \xleftarrow{\$} \mathbb{Z}_p^*$ and compute
$r \leftarrow (t^{k''} \bmod p) \bmod q$,
until $\omega \equiv r \bmod 2$.

k'' ←

Description

Extract

- To extract the embedded message ω compute $\omega \leftarrow r \bmod 2$.

Security

- To achieve indistinguishablility from Simmons' protocol, *Walter* must use sufficient parallel computing power.
- The more power *Walter* has, the longer the conveyed message can be.
- If *Walter* uses α *CU*, then the probability of *Walter* transmitting his message undetected is $1 - 1/2^{\alpha}$.
- The cuckoo's channel presented preserves the distribution of r.

Description

Public Parameters' Generation

- Select a prime number $q \geq 2^{\kappa}$.
- Select a prime number $p \geq 2^{\lambda}$ such that $q|p-1$.
- Choose an element $g \in \mathbb{Z}_p$ of order q.
- Choose two hash functions $h : \{0,1\}^* \rightarrow \mathbb{G}$ and $h' : \{0,1\}^* \times \mathbb{G} \rightarrow \mathbb{Z}_q^*$.
- Output the public parameters $pp = (p, q, g, h, h')$.

Description

Warden's Key Generation

- Choose $t \xleftarrow{\$} \mathbb{Z}_q^*$.
- Compute $z \leftarrow g^t$.
- Output the public key $pk_w = z$ and the secret key $sk_w = t$.

Description

Signer's Key Generation

- Choose $x \xleftarrow{\$} \mathbb{Z}_q^*$.
- Compute $y \leftarrow z^x$.
- Output the public key $pk = y$ and the secret key $sk = x$.

Description

Alice		*Walter*
	step ①	
Send a request (*req*) stating "I want to sign a message."	$\xrightarrow{req}$	
	step ②	
		Choose $a, k' \xleftarrow{\$} \mathbb{Z}_q^*$. Compute $\alpha \leftarrow g^{ak'} \bmod p$.
	$\xleftarrow{\alpha}$	
	step ③	
Choose $k \xleftarrow{\$} \mathbb{Z}_q^*$. Compute $h_0 \leftarrow h(m)$ and $\beta \leftarrow \alpha^{kh_0} \bmod p$.		
	$\xrightarrow{(h_0, \beta)}$	
	step ④	

Description

Alice		*Walter*
		Compute the following $\phi \leftarrow a^{-1} \bmod q$, $r \leftarrow \beta^{\phi} \bmod p$, $\epsilon \leftarrow k'^{-1} \bmod q$, $\gamma \leftarrow y^{\epsilon} \bmod p$.
	$\xleftarrow{(r,\gamma)}$	
	step ⑤	
Compute the following $e \leftarrow h'(m\|\|r)$, $f \leftarrow e^{x} \bmod p$, $\delta \leftarrow g^{kh_0} \bmod p$. Prepare the proof $\mathcal{P}$.		
	$\xrightarrow{(e,f,\delta,\mathcal{P})}$	
	step ⑥	
		If $\mathcal{P}$ is not valid `abort`. Compute $\eta \leftarrow k'(\gamma f \delta)^{-1} \bmod p$ and $\theta \leftarrow \eta^{-1} t \bmod p$.
	$\xleftarrow{\theta}$	
	step ⑦	

Description

Alice		*Walter*
	step ⑦	
Compute $s' \leftarrow kh_0 + \theta xe(\gamma f \delta)^{-1} \bmod q$.		
	$\xrightarrow{(m, s')}$	
	step ⑧	
		If $h_0 \neq h(m)$ or $e \neq h'(m \Vert r)$ then `abort`. Else compute $s \leftarrow k's' \bmod q$.
	$\xleftarrow{s}$	

Description

Verification

- Compute $r \leftarrow g^s y^{-e} \bmod p$ and $u \leftarrow h'(m \| r)$.
- Output `true` if and only if $u = e$. Else output `false`.

Description

Alice *Walter*

step ⑤

If $\omega \not\equiv r \bmod 2$ `abort`.
Compute the following
$e \leftarrow h(m\|r)$, $f \leftarrow e^x \bmod p$,
$\delta \leftarrow g^{kh_0} \bmod p$.
Prepare the proof $\mathcal{P}$.

$\xrightarrow{(e,f,\delta,\mathcal{P})}$

Description

Extract

- Compute $r \leftarrow g^s y^{-e} \bmod p$.
- To extract the embedded message ω compute $\omega \leftarrow r \bmod 2$.

Description

Alice *Walter*

step ④

Compute the following
$\phi \leftarrow a^{-1} \bmod q$, $r \leftarrow \beta^{\phi} \bmod p$,
$\epsilon \leftarrow k'^{-1} \bmod q$, $\gamma \leftarrow y^{\epsilon} \bmod p$.

$\xleftarrow{(r,\gamma)}$

step ⑥

If $\mathcal{P}$ is not valid `abort`.
Compute $\eta \leftarrow k'(\gamma f \delta)^{-1} \bmod p$ and
$\theta \leftarrow \eta^{-1} t \bmod p$.

$\xleftarrow{\theta}$

step ⑧

If $h_0 \neq h(m)$ or $e \neq h'(m \| r)$
then `abort`.
Else compute $s \leftarrow k's' \bmod q$.

$\xleftarrow{s}$

Description

Alice		*Walter*
	step ④	
		Choose $\phi \xleftarrow{\$} \mathbb{Z}_q^*$ and compute $r \leftarrow \beta^{\phi} \bmod p$, until $\omega \equiv r \bmod 2$. Denote by $k'' \leftarrow ak'\phi \bmod q$. Compute $\epsilon \leftarrow k''^{-1} \bmod q$ and $\gamma \leftarrow y^{\epsilon} \bmod p$.
	$\xleftarrow{(r,\gamma)}$	
	step ⑥	
		If $\mathcal{P}$ is not valid `abort`. Compute $\eta \leftarrow k''(\gamma f \delta)^{-1} \bmod p$ and $\theta \leftarrow \eta^{-1} t \bmod p$.
	$\xleftarrow{\theta}$	
	step ⑧	
		If $h_0 \neq h(m)$ or $e \neq h'(m\|r)$ `abort`. Compute $s \leftarrow k''s' \bmod q$.
	$\xleftarrow{s}$	

Description

Extract

- Compute $r \leftarrow g^s y^{-e} \bmod p$.
- To extract the embedded message ω compute $\omega \leftarrow r \bmod 2$.

Security

- The *Verification* algorithm outputs `true` if all the steps are followed.
- According to Zhang *et al.*, *Walter* will not deviate from the signing protocol.
- Thus, in Step 4, *Walter* has to supply *Alice* with (r, γ), θ and s of a given distribution.
- The cuckoo's channel preserves the distributions of (r, γ), θ and s.

Description

Public Parameters' Generation

- Select an elliptic curve $E(\mathbb{Z}_p)$ defined over $\mathbb{Z}_p$, where p is prime.
- Generate a prime number $q \geq 2^\lambda$, such that q divides $|E(\mathbb{Z}_p)|$.
- Generate a point $P \in E(\mathbb{Z}_p)$ of order q.
- Select a hash function $h : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$.
- Output the public parameters $pp = (q, P, E(\mathbb{Z}_p), h)$.

Description

Signer's Key Generation

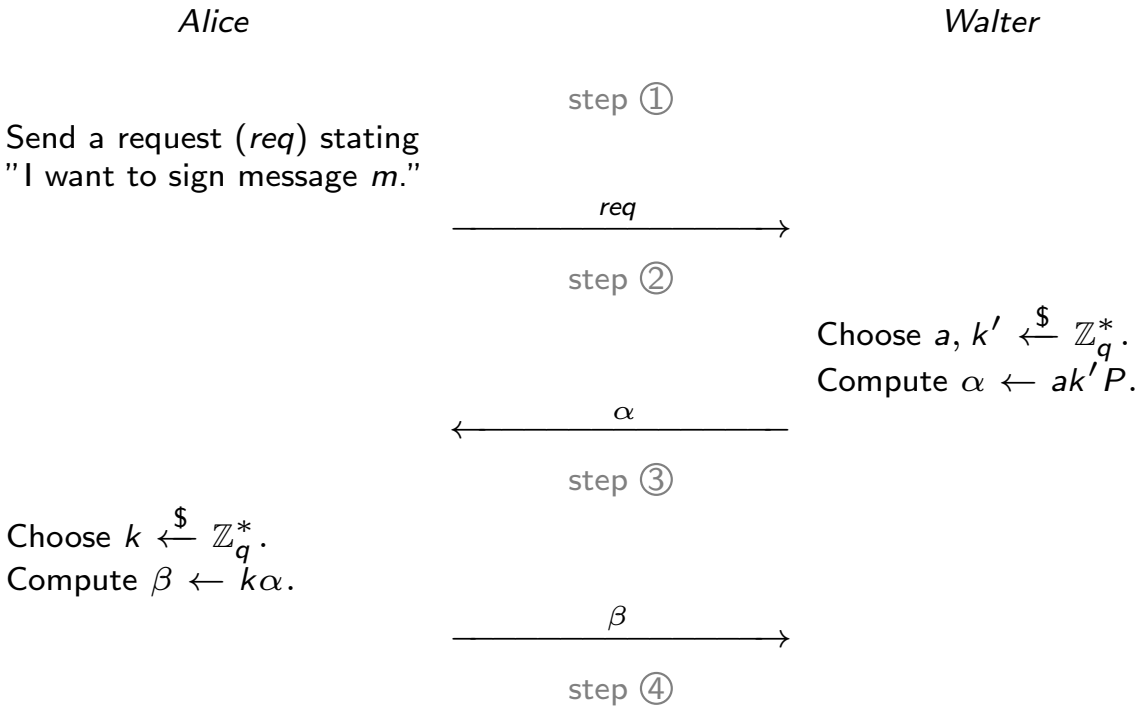
- Choose $d \xleftarrow{\$} \mathbb{Z}_q^*$.
- Compute $Q \leftarrow dP$.
- Output the public key $pk = Q$ and the secret key $sk = d$.

Description

Warden's Key Generation

- Choose $t \xleftarrow{\$} \mathbb{Z}_q^*$.
- Compute $T \leftarrow tQ = (x_t, y_t)$.
- Set $h_t = h(x_t \| y_t)$.
- Output the public key $pk_w = h_t$ and the secret key $sk_w = t$.

Description



Description

Alice		*Walter*
		Compute the following $(x_0, y_0) \leftarrow a^{-1}\beta$, $r \leftarrow x_0 \bmod q$, $\theta \leftarrow r^{-1}t^{-1} \bmod q$.
	$\xleftarrow{\theta}$	
	step ⑤	
Compute $s' \leftarrow k^{-1}[h(m)\theta + d] \bmod q$.		
	$\xrightarrow{s'}$	
	step ⑥	
		Compute $s \leftarrow k'^{-1}s'\theta^{-1} \bmod q$. If (r, s, T) is not valid `abort`.
	$\xleftarrow{(r,s,T)}$	

Description

Verification

- Compute $u_1 \leftarrow h(m)s^{-1} \bmod q$, $u_2 \leftarrow rs^{-1} \bmod q$ and $h_t^* = h(x_t \| y_t)$.
- Output `true` if and only if $v = r$ and $h_t^* = h_t$. Otherwise, output `false`.

Description

Alice *Walter*

step ⑤

Compute $(x_s, y_s) \leftarrow \theta^{-1}Q$.
If $\omega \not\equiv x_s \bmod 2$ `abort`.
Compute
$s' \leftarrow k^{-1}[h(m)\theta + d] \bmod q$.

$\xrightarrow{\quad s' \quad}$

Description

Extract

- Compute $rT = (x_s, y_s)$.
- To extract the embedded message ω compute $\omega \leftarrow x_s \bmod 2$.

Description

step ④

Compute the following
$(x_0, y_0) \leftarrow a^{-1}\beta$, $r \leftarrow x_0 \bmod q$,
$\theta \leftarrow r^{-1}t^{-1} \bmod q$.

θ ←

step ⑥

Compute $s \leftarrow k'^{-1}s'\theta^{-1} \bmod q$.
If (r, s, T) is not valid `abort`.

Description

step ④

Choose $\phi \xleftarrow{\$} \mathbb{Z}_q^*$ and compute
$(x_0, y_0) \leftarrow \phi\beta$, $r \leftarrow x_0 \bmod q$,
until $\omega \equiv r \bmod 2$.
Compute $\theta \leftarrow r^{-1}t^{-1}$.

θ ←

step ⑥

Denote by $k'' \leftarrow ak'\phi \bmod q$.
Compute $s \leftarrow k''^{-1}s'\theta^{-1} \bmod q$.
If (r, s, T) is not valid `abort`.

(r,s,T) ←

Description

Extract

- To extract the embedded message compute $\omega \leftarrow r \bmod 2$.

Security

- The *Verification* algorithm outputs `true` if all the steps are followed.
- The cuckoo's channel preserves the distributions of θ and (r, s).

Conclusions

- Zhang *et al.* and Dong *et al.* propose two signature protocols that they claim to be subliminal-free.
- We have proved that their claims are false.
- Since, the main utility of these protocols was to be subliminal-free and they failed to be so, we suggest that users employ other means of protection against subliminal channels with a lower communication overhead.

Questions?