Modern Cryptography

Public Key Systems for Secret Sharing

Graeme Taylor

Edinburgh

October 2008

What is Cryptography?

Cryptography

From Greek *kryptos* - hidden - and *graphos* - writing - cryptography is the use of codes to disguise messages.

The main challenge in cryptography

Cryptography

How can you communicate securely over an insecure channel?

Classical Cryptography

The encoding and decoding of messages is at least 2,000 years old- in Roman times, the *Caesar shift cipher* was employed.

Classical Cryptography

The encoding and decoding of messages is at least 2,000 years old- in Roman times, the *Caesar shift cipher* was employed.
For instance, with a shift of 4:

Plain	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Cipher	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D

Classical Cryptography

The encoding and decoding of messages is at least 2,000 years old- in Roman times, the *Caesar shift cipher* was employed.
For instance, with a shift of 4:

Plain	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Cipher	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D

The word "cryptography" would thus become "GVCTSKVETLC". The shift number 4 is the "key" to both locking and unlocking the *enciphered* message (symmetric encryption).

Classical Cryptography

Although the encryption/decryption systems became more sophisticated, until the 20th century, the basic idea remained the same:

Classical Cryptography

Although the encryption/decryption systems became more sophisticated, until the 20th century, the basic idea remained the same:

- The sender converts the message into *ciphertext* using an encryption system.

$$\text{Secret Key} + \text{Plaintext} \longrightarrow \text{Ciphertext}$$

Classical Cryptography

Although the encryption/decryption systems became more sophisticated, until the 20th century, the basic idea remained the same:

- The sender converts the message into *ciphertext* using an encryption system.

$$\text{Secret Key} + \text{Plaintext} \longrightarrow \text{Ciphertext}$$

- The receiver converts the ciphertext back into *plaintext* using a corresponding decryption system.

$$\text{Secret Key} + \text{Ciphertext} \longrightarrow \text{Plaintext}$$

Classical Cryptography

Problems with classical cryptography

- If an adversary learns the decryption key and system, they can decipher messages, and thus secrecy is lost.
- If an adversary learns the encryption key and system, they can encipher messages, and thus trust is lost.

Classical Cryptography

The biggest problem with private key cryptography

In order to share secrets, you must first have shared a secret!

Public Key Cryptography

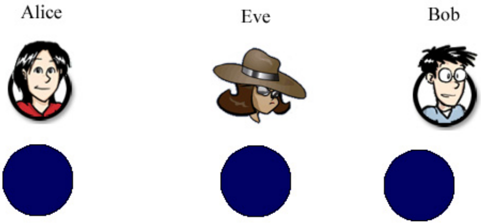
The challenge for modern cryptography

Can you establish a secret with a previously uncontacted stranger, without sharing the same secret with anyone listening in?

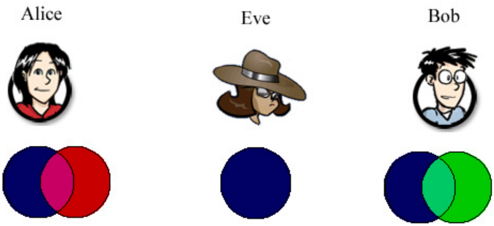
Secret sharing with paint

Secret sharing with paint

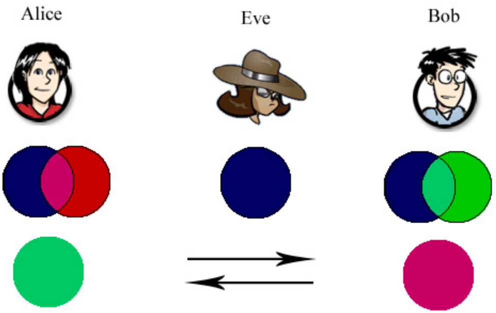
Secret sharing with paint



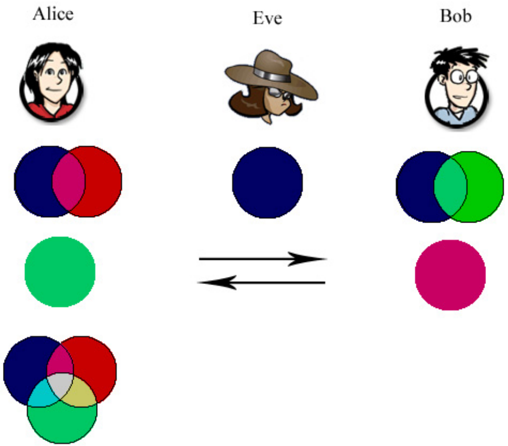
Secret sharing with paint



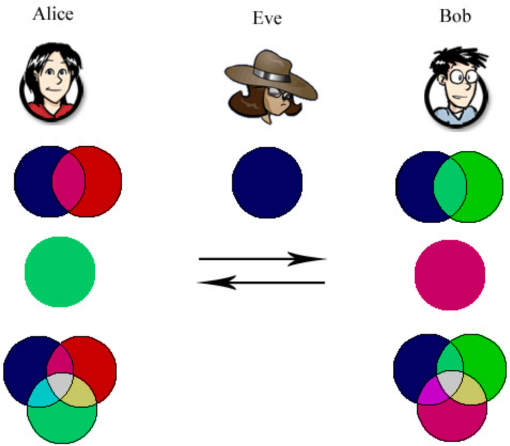
Secret sharing with paint



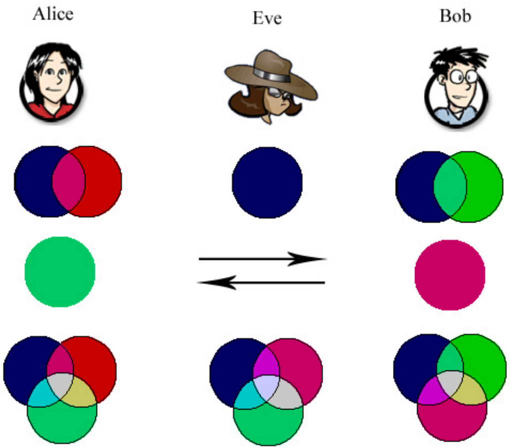
Secret sharing with paint



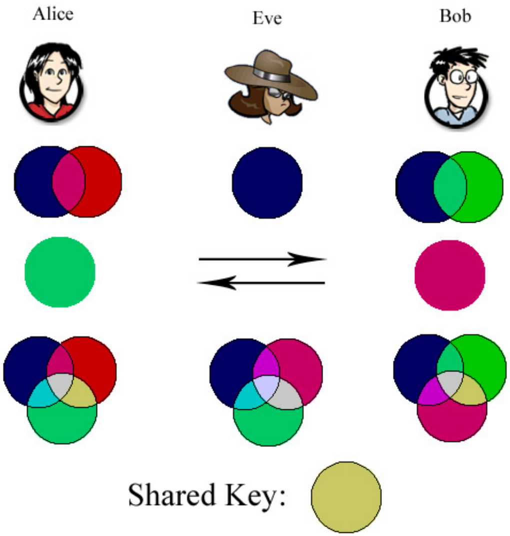
Secret sharing with paint



Secret sharing with paint



Secret sharing with paint



Secret sharing with paint

- The public information is not enough to learn the shared secret, so it really is a secret unless you know a private colour.

Secret sharing with paint

- The public information is not enough to learn the shared secret, so it really is a secret unless you know a private colour.
- But you can't unmix paint, so Eve can't learn a private colour from the mix and the base.

Secret sharing with paint

- The public information is not enough to learn the shared secret, so it really is a secret unless you know a private colour.
- But you can't unmix paint, so Eve can't learn a private colour from the mix and the base.
- So Eve can't learn the shared secret.

Secret sharing with paint

- The public information is not enough to learn the shared secret, so it really is a secret unless you know a private colour.
- But you can't unmix paint, so Eve can't learn a private colour from the mix and the base.
- So Eve can't learn the shared secret.
- The order in which paint is mixed does not matter- so Alice and Bob reach the same secret result.

Secret sharing with mathematics

Can we mimic these properties mathematically?

Secret sharing with mathematics

Can we mimic these properties mathematically?

Definition

An injective function is described as a *one-way function* if, like mixing paint, it's easy to compute the output from the inputs, but (practically) impossible to compute the inputs from the output.

Secret sharing with mathematics

Can we mimic these properties mathematically?

Definition

An injective function is described as a *one-way function* if, like mixing paint, it's easy to compute the output from the inputs, but (practically) impossible to compute the inputs from the output.

Problem

No one has managed to prove that a one-way function really exists!

A possible one-way function

Let $(G, \oplus)$ be a finite additive group of order N .

A possible one-way function

Let $(G, \oplus)$ be a finite additive group of order N .

Definition

The scalar multiple $[t]g$ of $g \in G$ is

$$\underbrace{g \oplus g \oplus \cdots \oplus g}_{t \text{ copies}}$$

A possible one-way function

Let $(G, \oplus)$ be a finite additive group of order N .

Definition

The scalar multiple $[t]g$ of $g \in G$ is

$$\underbrace{g \oplus g \oplus \cdots \oplus g}_{t \text{ copies}}$$

So we can consider the map

$$f : \mathbb{Z}/N\,\mathbb{Z} \times G \to G$$

$$f(n, g) = [n]g$$

Computing f

Given t, g we can compute $h = [t]g$ in $O(log_2(t))$ group operations by a fast exponentiation algorithm.

Computing f

Given t, g we can compute $h = [t]g$ in $O(log_2(t))$ group operations by a fast exponentiation algorithm.

Example (Binary Double-and-add)

Let t have binary digits $d_k d_{k-1} d_{k-2} \ldots d_0$. Set $T = g$.

Computing f

Given t, g we can compute $h = [t]g$ in $O(log_2(t))$ group operations by a fast exponentiation algorithm.

Example (Binary Double-and-add)

Let t have binary digits $d_k d_{k-1} d_{k-2} \ldots d_0$. Set $T = g$.
For i from $k-1$ to 0,

If d_i=0, set $T = T \oplus T$.
Else, set $T = T \oplus T \oplus g$.

Computing f

Given t, g we can compute $h = [t]g$ in $O(log_2(t))$ group operations by a fast exponentiation algorithm.

Example (Binary Double-and-add)

Let t have binary digits $d_k d_{k-1} d_{k-2} \ldots d_0$. Set $T = g$.
For i from $k-1$ to 0,
If d_i=0, set $T = T \oplus T$.
Else, set $T = T \oplus T \oplus g$.
Then $T = [t]g = h$ as required.

Computing f

Example (t=83)

$83 = (1010011)_2$ so our sequence is

$$T = g$$

Computing f

Example (t=83)

$83 = (1010011)_2$ so our sequence is

$$\begin{aligned} & T = g \\ d_5 = 0, \text{double} \quad & T = T \oplus T = [2]g \end{aligned}$$

Computing f

Example (t=83)

$83 = (1010011)_2$ so our sequence is

$$\begin{aligned} & & T &= g \\ d_5 &= 0, \text{double} & T &= T \oplus T = [2]g \\ d_4 &= 1, \text{double-and-add} & T &= T \oplus T \oplus g = [5]g \end{aligned}$$

Computing f

Example (t=83)

$83 = (1010011)_2$ so our sequence is

$$\begin{aligned} & T = g \\ d_5 = 0, \text{double} \quad & T = T \oplus T = [2]g \\ d_4 = 1, \text{double-and-add} \quad & T = T \oplus T \oplus g = [5]g \\ d_3 = 0, \text{double} \quad & T = T \oplus T = [10]g \end{aligned}$$

Computing f

Example (t=83)

$83 = (1010011)_2$ so our sequence is

$$\begin{aligned} & & T &= g \\ d_5 &= 0, \text{double} & T &= T \oplus T = [2]g \\ d_4 &= 1, \text{double-and-add} & T &= T \oplus T \oplus g = [5]g \\ d_3 &= 0, \text{double} & T &= T \oplus T = [10]g \\ d_2 &= 0, \text{double} & T &= T \oplus T = [20]g \end{aligned}$$

Computing f

Example (t=83)

$83 = (1010011)_2$ so our sequence is

$$\begin{aligned}
& & T &= g \\
d_5 &= 0, \text{double} & T &= T \oplus T = [2]g \\
d_4 &= 1, \text{double-and-add} & T &= T \oplus T \oplus g = [5]g \\
d_3 &= 0, \text{double} & T &= T \oplus T = [10]g \\
d_2 &= 0, \text{double} & T &= T \oplus T = [20]g \\
d_1 &= 1, \text{double-and-add} & T &= T \oplus T \oplus g = [41]g
\end{aligned}$$

Computing f

Example (t=83)

$83 = (1010011)_2$ so our sequence is

$$
\begin{aligned}
& & T &= g \\
d_5 &= 0, \text{double} & T &= T \oplus T = [2]g \\
d_4 &= 1, \text{double-and-add} & T &= T \oplus T \oplus g = [5]g \\
d_3 &= 0, \text{double} & T &= T \oplus T = [10]g \\
d_2 &= 0, \text{double} & T &= T \oplus T = [20]g \\
d_1 &= 1, \text{double-and-add} & T &= T \oplus T \oplus g = [41]g \\
d_0 &= 1, \text{double-and-add} & T &= T \oplus T \oplus g = [83]g
\end{aligned}
$$

Computing f^{-1}

The reverse of scalar multiplication is the *Discrete Logarithm Problem*.

Definition (DLP)

Given $g, h \in G$, find t such that $[t]g = h$.

Computing f^{-1}

The reverse of scalar multiplication is the *Discrete Logarithm Problem*.

Definition (DLP)

Given $g, h \in G$, find t such that $[t]g = h$.

Theorem (Shoup, '97)

If $\mathcal{A}$ is an algorithm that reads in g, h, performs m group operations and then returns an answer $v \in \mathbb{Z}/N\mathbb{Z}$, then the probability that $t = v$ is $O(m^2/p)$, for p the largest prime dividing N.

Computing f^{-1}

The reverse of scalar multiplication is the *Discrete Logarithm Problem*.

Definition (DLP)

Given $g, h \in G$, find t such that $[t]g = h$.

Theorem (Shoup, '97)

If $\mathcal{A}$ is an algorithm that reads in g, h, performs m group operations and then returns an answer $v \in \mathbb{Z}/N\mathbb{Z}$, then the probability that $t = v$ is $O(m^2/p)$, for p the largest prime dividing N.

So for a non-negligible probability of success, $\mathcal{A}$ must perform $O(\sqrt{p})$ group operations.

DLP Cryptography

If G is a group of prime order p then

- Scalar multiplication takes $O(k)$ group operations for $k = log_2(p)$.

DLP Cryptography

If G is a group of prime order p then

- Scalar multiplication takes $O(k)$ group operations for $k = log_2(p)$.
- DLP takes $O(\sqrt{p}) = O(2^{k/2})$ operations- exponentially harder!

DLP Cryptography

If G is a group of prime order p then

- Scalar multiplication takes $O(k)$ group operations for $k = log_2(p)$.
- DLP takes $O(\sqrt{p}) = O(2^{k/2})$ operations- exponentially harder!
- The order of scalar multiplications doesn't matter:
  $[a]([b]g) = [ab]g = [ba]g = [b]([a]g)$.

DLP Cryptography

If G is a group of prime order p then

- Scalar multiplication takes $O(k)$ group operations for $k = log_2(p)$.
- DLP takes $O(\sqrt{p}) = O(2^{k/2})$ operations- exponentially harder!
- The order of scalar multiplications doesn't matter: $[a]([b]g) = [ab]g = [ba]g = [b]([a]g)$.
- So we can use prime groups to securely generate shared secrets.

Except... which group?

Shoup's result assumes no knowledge of the underlying group. But any implementation requires a group to be chosen, and this may introduce additional structure that makes the DLP easier.

Except... which group?

Shoup's result assumes no knowledge of the underlying group. But any implementation requires a group to be chosen, and this may introduce additional structure that makes the DLP easier.

Example

For $\mathbb{Z}/p\mathbb{Z}$ with addition modulo p, the DLP is very easy! Just use Euclid's algorithm.

Except... which group?

Shoup's result assumes no knowledge of the underlying group. But any implementation requires a group to be chosen, and this may introduce additional structure that makes the DLP easier.

Example

For $\mathbb{Z}/p\mathbb{Z}$ with addition modulo p, the DLP is very easy! Just use Euclid's algorithm.

Currently fashionable choice is the group of rational points of an elliptic curve over a finite field, since there is no obvious reduction to $\mathbb{Z}/p\mathbb{Z}$.

Limitations and open problems

- We can't prove that there are *any* one-way functions.

Limitations and open problems

- We can't prove that there are *any* one-way functions.
- If there are, ECDLP might not be one of them.

Limitations and open problems

- We can't prove that there are *any* one-way functions.
- If there are, ECDLP might not be one of them.
- Even with perfect one-way functions, the protocol might be flawed.

Limitations and open problems

- We can't prove that there are *any* one-way functions.
- If there are, ECDLP might not be one of them.
- Even with perfect one-way functions, the protocol might be flawed.
- Even with perfect crypto and perfect protocols, implementation may disclose secrets.

Limitations and open problems

- We can't prove that there are *any* one-way functions.
- If there are, ECDLP might not be one of them.
- Even with perfect one-way functions, the protocol might be flawed.
- Even with perfect crypto and perfect protocols, implementation may disclose secrets.
- After establishing a secret, we need a classical cryptosystem that's at least as secure.

XLEROCSY JSV PMWXIRMRK!

THANKYOU FOR LISTENING!