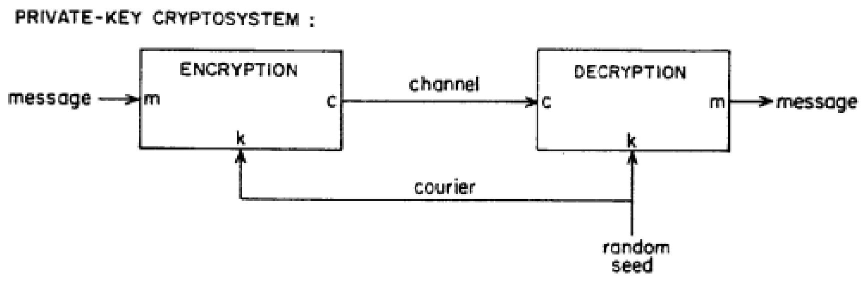
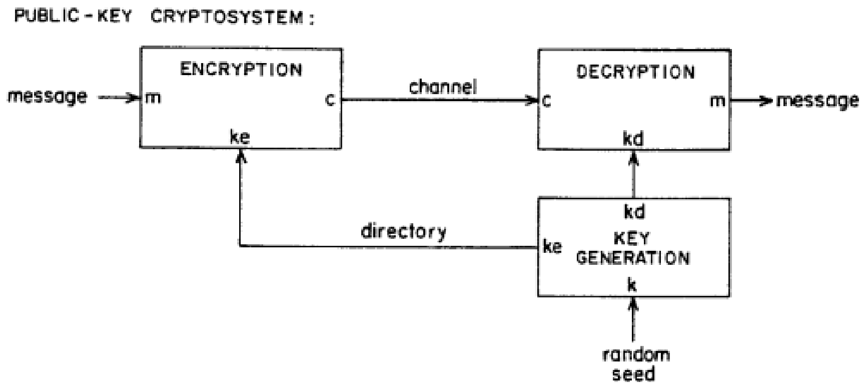
BASICS ON
IDENTITY BASED CRYPTOGRAPHY

By Harsh Raj Sahu from the paper of Adi Shamir

Before we start our journey in the field of identity-based Cryptography, we should have a general idea about public key Cryptography and private key cryptography. The main basis of private key cryptography is that a single key is used to encrypt and decrypt the plain text and cipher text, as it is a part of Symmetric Cryptosystem. In the case of public key cryptography, we use two different keys first one is a public key and second one is a private key, basically we encrypt the message with receiver's public key and receiver decrypts the message with his private key.

from Adi Shamir's paper





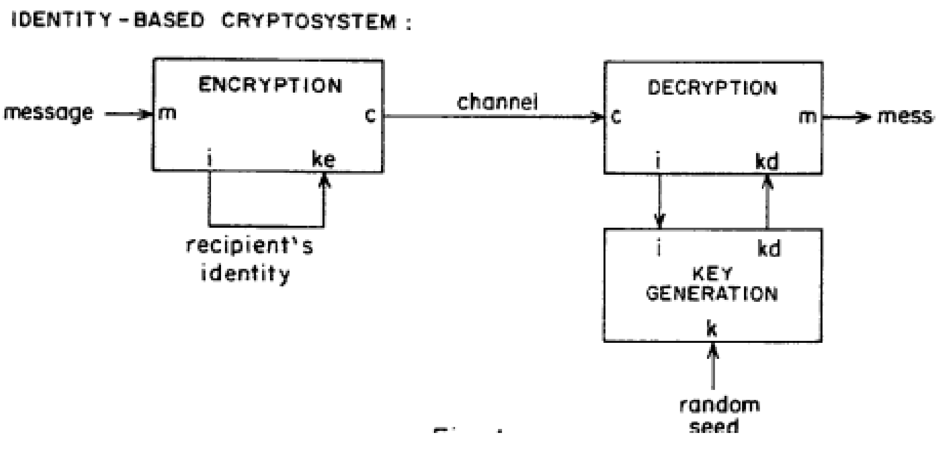
So, the public key cryptography is world recognised method for cryptography but there are few problems with certificates and certification authority, so Adi Shamir proposed a method for the following problems. It is known as Identity based Cryptography or IBC. The IBC removes the need for certificates to overcome the drawback of certificate generation and certificate management in public-key cryptography.

INTRODUCTION:

In identity-based cryptosystems (IBC), identity is also the public key. This has two clear advantages over traditional public-key cryptosystems. First, certificates are not needed to bind the two independent information units, identity and public key. Second, key management is simplified because users can easily remember public keys for identities such as email addresses or domain names. There exists a wide variety of IBCs with widely differing properties and application domains. However, since most of them need a trusted third party that can derive the private keys of participants, those systems are not suitable for applications where this is a problem. This might contribute to the fact that, in practice, applications of IBC are still relatively rare, although there are various standards for IBC. As Adi Shamir proposed in his paper:

IDEA:

The scheme assumes the existence of trusted key generation centres, whose sole purpose is to give each user a personalized smart card when he first joins the network. The information embedded in this card enables the user to sign and encrypt the messages he sends and to decrypt and verify the messages he receives in a totally independent way, regardless of the identity of the other party. Here the users just use their ID as their public keys to encrypt the message and send it by the open server. Any combination of name, social security number, street address, office number or telephone number can be used (depending on the context) provided that it uniquely identifies the user in a way he cannot later deny, so this is the overall idea behind Identity Based Cryptography.



From Adi Shamir

Ke and Kd are the encryption and decryption key in the following IBC based figure so here we can see the message is being encrypted by the identity of the sender and it moves in the open channel to the receiver, when receiver receives the message it sends the identity to the public key generator (PKG) and the PKG sends a decryption key to the receiver for the decryption of the message.

Definition:

In identity-based cryptosystems, the public key is also the identity. This has two clear advantages over traditional public-key cryptosystems: first, there is no need for certificates to bind the two independent information units, identity and public key, and second, key management is simplified because users can easily remember public keys, at least for identities such as email addresses or domain names. These two properties were the primary motivation of Adi Shamir when he introduced this type of cryptography in 1984.

An Identity-based encryption scheme can be specified by four randomized Algorithms:

1. **Setup**: takes a security parameter k and returns params (System parameters) and master key. The System parameter include a description of finite message space M1 and a description of finite cipher text space C. Intuitively the system parameters will be publicly known, while the master key will only be known to Private Key Generator.

2. **Extract:** takes as input params, master key, an ID belongs to {0,1} * , and returns a private key d, here ID is an Arbitrary string that will be used as a private key and d is the corresponding private decryption key, the extract algorithm extracts a private key from the given public key.

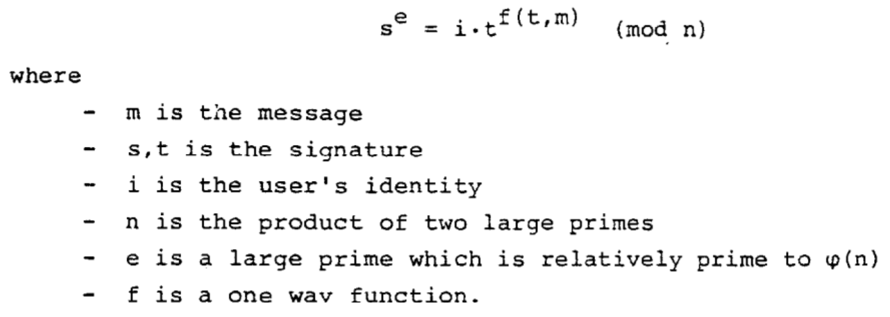
3. **Encrypt:** takes an input params ID and m belongs to M it returns a cipher text c belongs to C.

4. **Decrypt:** takes as input params and c belongs to C and a private key d, and it returns m belongs to M.

The Implementation:

TO implement the idea described in the previous section, we need a public-key scheme with two additional properties: (a) When the seed k is known, secret keys can be easily computed for a non-negligible fraction of the possible public keys. (b) The problem of computing the seed k from specific public/secret key pairs generated with this k is intractable.

The signature scheme is based on the verification condition:



$$s^e = i \cdot t^{f(t,m)} \pmod n$$

where

- m is the message
- s,t is the signature
- i is the user's identity
- n is the product of two large primes
- e is a large prime which is relatively prime to $\varphi(n)$
- f is a one wav function.

Adi Shamir implementation

The parameters n, e and the function f are chosen by the key generation centre, and all the users have the same n, e and the same algorithmic description of f stored in their smart cards. These values can be made public, but the factorization of n should be known only to the key generation centre. The only difference between users is the value of i, and the secret key which corresponds to i is the (unique) number g such that:

$$g^e = i \pmod{n}.$$

This g can be easily computed by the key generation centre, but if the RSA scheme is secure no one else can extract e- th roots mod n. Each message m has a large number of possible (s, t) signatures, but their density is so low that a random search is extremely unlikely to discover any one of them. Any attempt to set one of (s, t) to a random value and solve for the other variable requires the extraction of modular roots, which is believed to be an exceedingly difficult computational task. However, when g is known, there is a very simple way to generate any number of signatures of any message even when the factorization of n is unknown. To sign the message m, the user chooses a random number r and computes

$$t = r^e \pmod{n}.$$

The verification condition can be rewritten as

$$s^e = g^e \cdot r^{ef(t,m)} \pmod{n}.$$

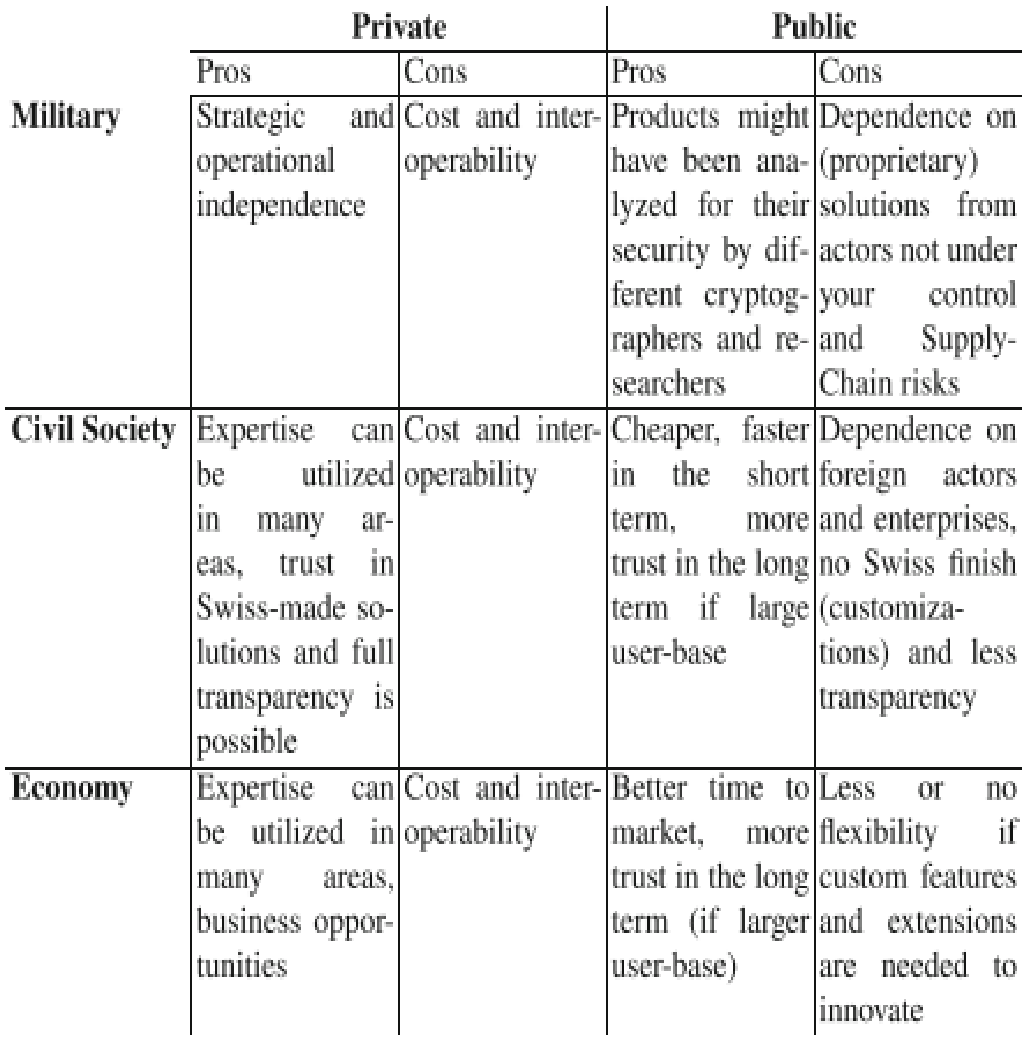
Since e is relatively prime to $\varphi(n)$, we can eliminate the common factor e from the exponents

$$s = g \cdot r^{f(t,m)} \pmod{n}$$

and thus, s can be computed without any root extraction.

TO prevent attacks based on multiplicative relationships between the identities (and thus also the g values) of different users, it is advisable to expand the string that describes the user's identity into a long pseudo-random string via a universal one-way function and use the expanded form as i in the verification condition. Since everyone in the network knows how to apply this function, the scheme retains its identity-based flavour even though the signature verification key is not strictly equal to the user's identity.

Different Field Implementation Possibilities:



	Private		Public	
	Pros	Cons	Pros	Cons
Military	Strategic and operational independence	Cost and inter-operability	Products might have been analyzed for their security by different cryptographers and researchers	Dependence on (proprietary) solutions from actors not under your control and Supply-Chain risks
Civil Society	Expertise can be utilized in many areas, trust in Swiss-made solutions and full transparency is possible	Cost and inter-operability	Cheaper, faster in the short term, more trust in the long term if large user-base	Dependence on foreign actors and enterprises, no Swiss finish (customizations) and less transparency
Economy	Expertise can be utilized in many areas, business opportunities	Cost and inter-operability	Better time to market, more trust in the long term (if larger user-base)	Less or no flexibility if custom features and extensions are needed to innovate

Conclusion:

Identity-based encryption systems are characterized by the fact that the public key is easy to remember, and the implementations skip the step of linking the public key to a specific identity. However, they also have some disadvantages. In particular, the trust that must be placed in the trusted third party is significantly higher than in traditional public key systems. Whether these and other potentially disadvantageous properties, such as the lack of well-tested quantum-safe solutions, are relevant depends on the specific use case. Moreover, the lively research activity in the field can potentially mitigate or eliminate undesirable properties.

References:

1. Adi Shamir's IDENTITY-BASED CRYPTOSYSTEMS AND SIGNATURE SCHEMES
2. Dan Boneh and Mathew Franklin's IDENTITY-BASED ENCRYPTION AND WELL-PAIRING
3. Valentin Mulder Alain Mermoud Vincent Lenders Bernhard Tellenbach Trends in Data Protection and Encryption Technologies book